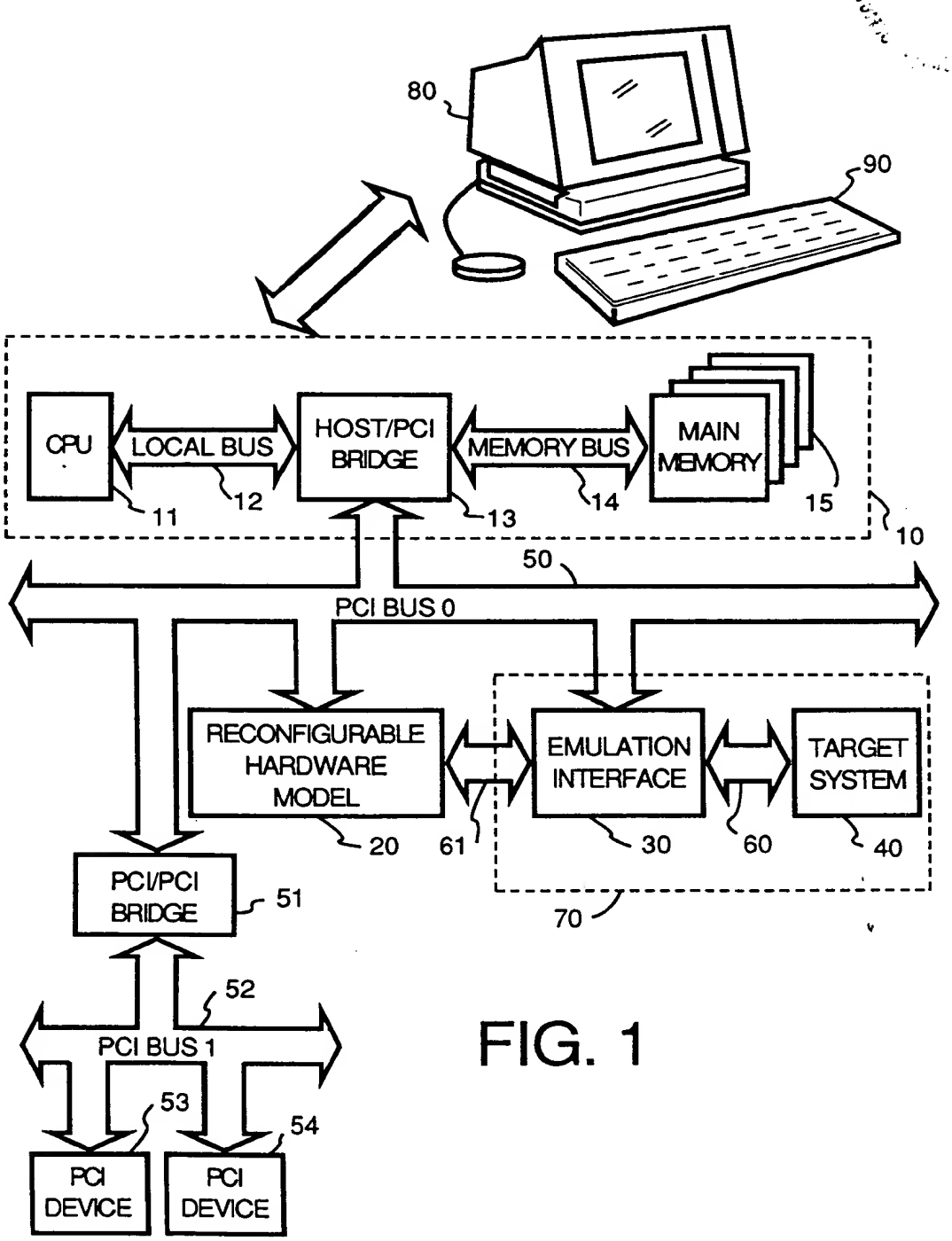


FIG. 1

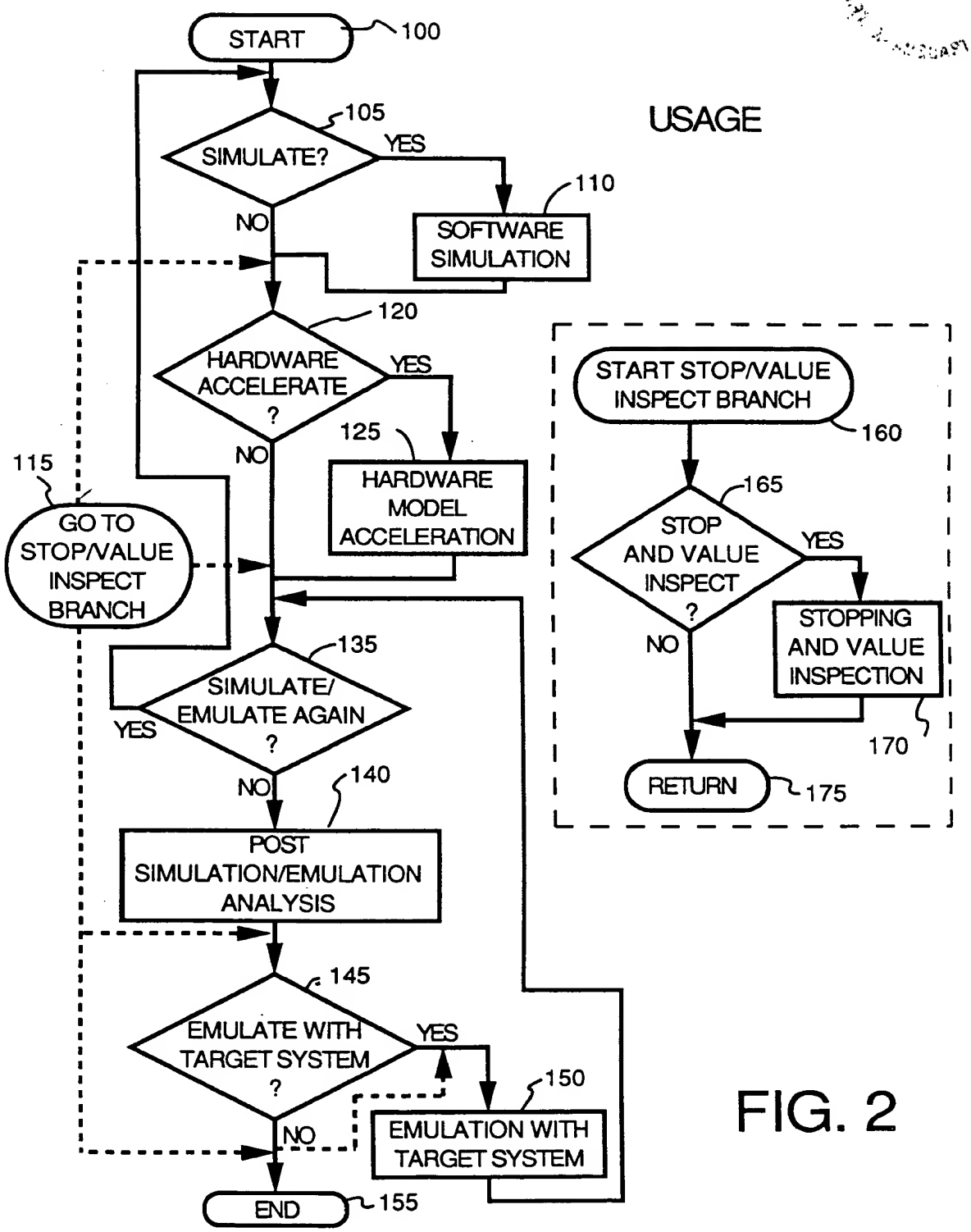

USAGE
START
100
105
SIMULATE?
YES
NO
110
SOFTWARE SIMULATION
120
HARDWARE ACCELERATE ?
YES
NO
125
HARDWARE MODEL ACCELERATION
115
GO TO STOP/VALUE INSPECT BRANCH
135
SIMULATE/ EMULATE AGAIN ?
YES
NO
140
POST SIMULATION/EMULATION ANALYSIS
145
EMULATE WITH TARGET SYSTEM ?
YES
NO
150
EMULATION WITH TARGET SYSTEM
END
155
START STOP/VALUE INSPECT BRANCH
160
165
STOP AND VALUE INSPECT ?
YES
NO
STOPPING AND VALUE INSPECTION
170
RETURN
175

FIG. 2

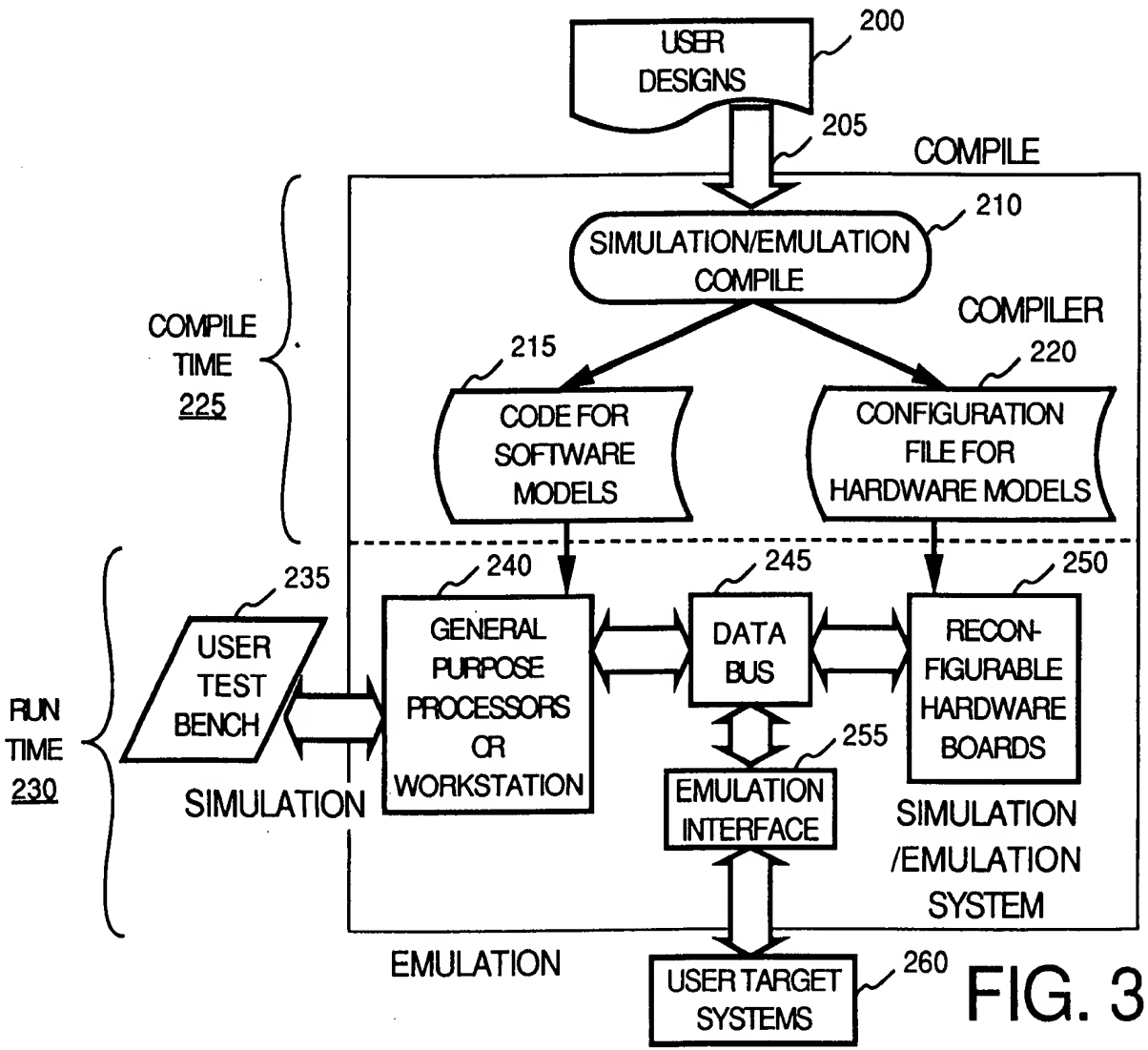


FIG. 3

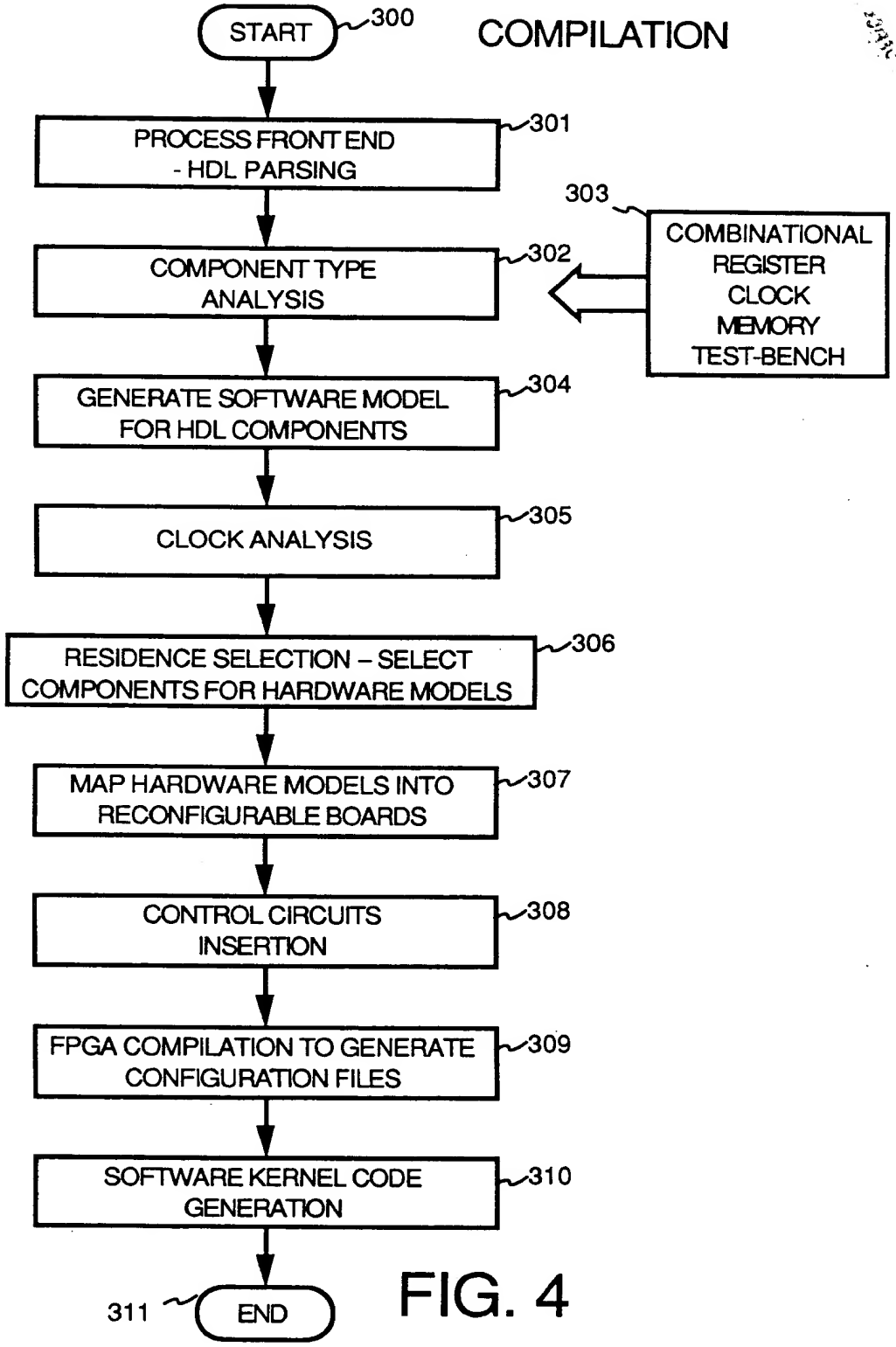

START
300
COMPILATION
PROCESS FRONT END
- HDL PARSING
301
303
COMBINATIONAL
REGISTER
CLOCK
MEMORY
TEST-BENCH
COMPONENT TYPE
ANALYSIS
302
GENERATE SOFTWARE MODEL
FOR HDL COMPONENTS
304
CLOCK ANALYSIS
305
RESIDENCE SELECTION – SELECT
COMPONENTS FOR HARDWARE MODELS
306
MAP HARDWARE MODELS INTO
RECONFIGURABLE BOARDS
307
CONTROL CIRCUITS
INSERTION
308
FPGA COMPILATION TO GENERATE
CONFIGURATION FILES
309
SOFTWARE KERNEL CODE
GENERATION
310
311
END

FIG. 4

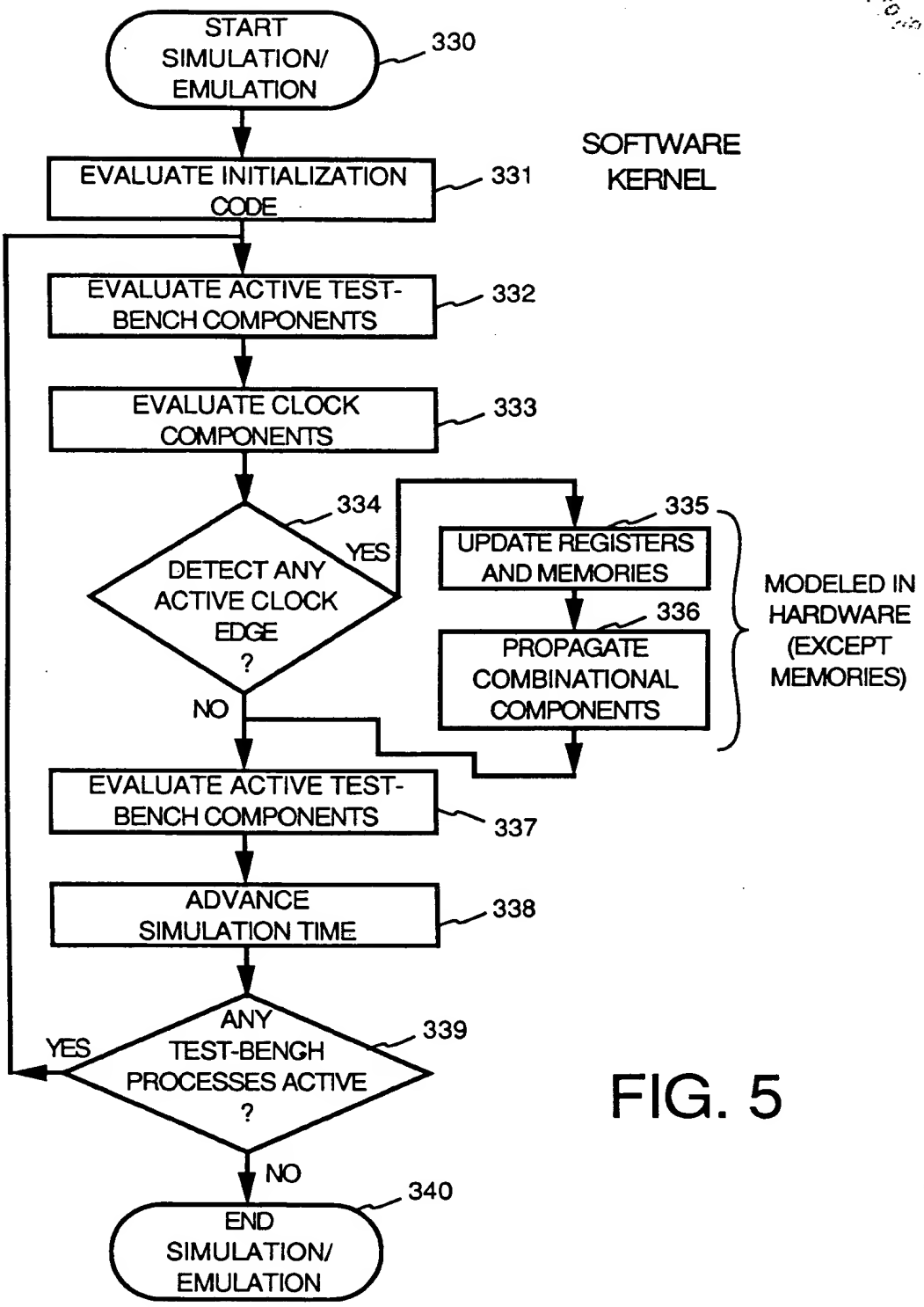

START SIMULATION/ EMULATION
330
EVALUATE INITIALIZATION CODE
331
SOFTWARE KERNEL
EVALUATE ACTIVE TEST-BENCH COMPONENTS
332
EVALUATE CLOCK COMPONENTS
333
334
DETECT ANY ACTIVE CLOCK EDGE ?
YES
NO
335
UPDATE REGISTERS AND MEMORIES
336
PROPAGATE COMBINATIONAL COMPONENTS
MODELED IN HARDWARE (EXCEPT MEMORIES)
EVALUATE ACTIVE TEST-BENCH COMPONENTS
337
ADVANCE SIMULATION TIME
338
ANY TEST-BENCH PROCESSES ACTIVE ?
339
YES
NO
END SIMULATION/ EMULATION
340

FIG. 5

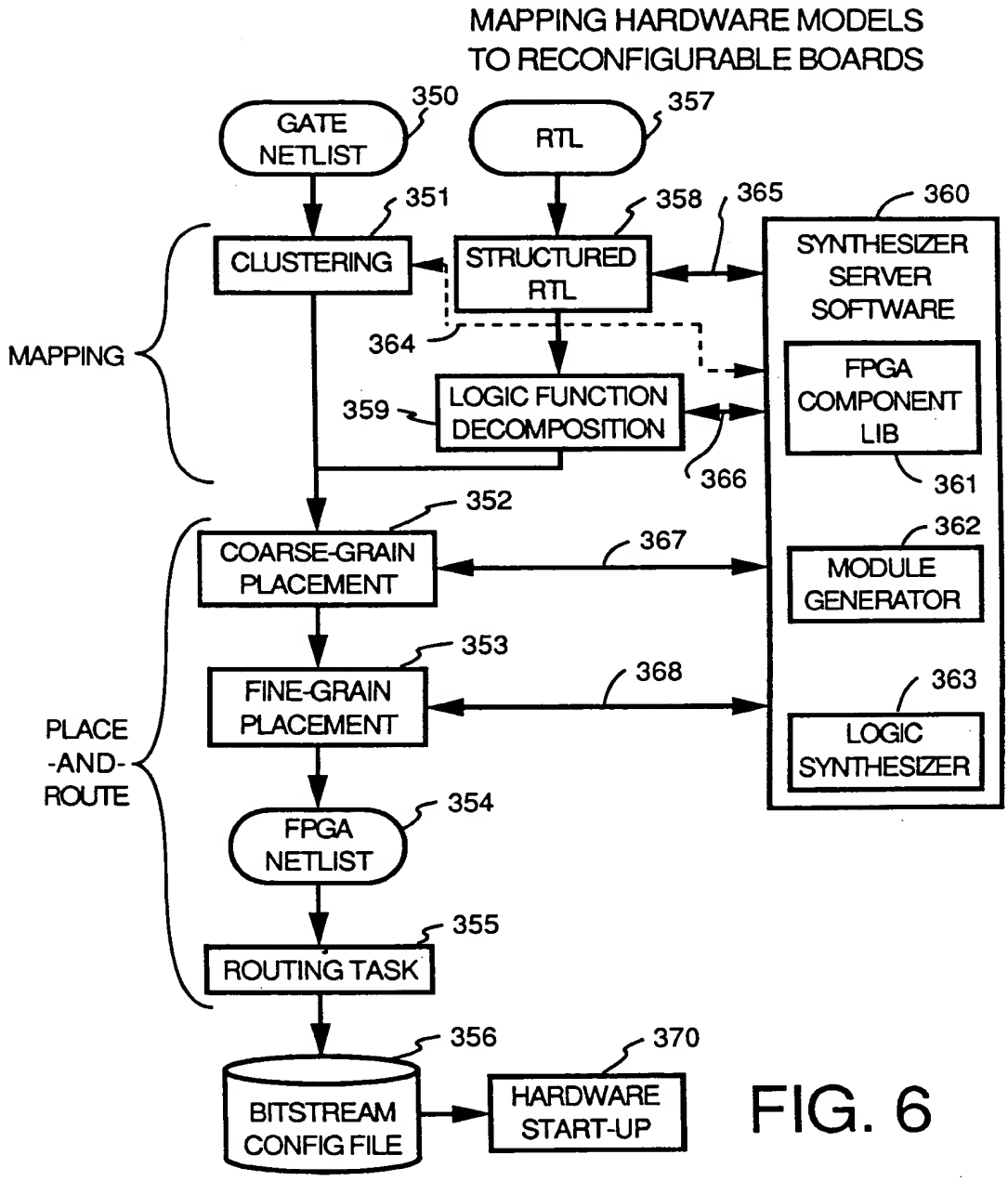


FIG. 6

	F11	F12	F13	F14	F21	F22	F23	F24	F31	F32	F33	F34	F41	F42	F43	F44
F11	1	1	1	1	1	0	0	0	1	0	0	0	1	0	0	0
F12	1	1	1	1	0	1	0	0	0	1	0	0	0	1	0	0
F13	1	1	1	1	0	0	1	0	0	0	1	0	0	0	1	0
F14	1	1	1	1	0	0	0	1	0	0	0	1	0	0	0	1
F21	1	0	0	0	1	1	1	1	1	0	0	0	1	0	0	0
F22	0	1	0	0	1	1	1	1	0	1	0	0	0	1	0	0
F23	0	0	1	0	1	1	1	1	0	0	1	0	0	0	1	0
F24	0	0	0	1	1	1	1	1	0	0	0	1	0	0	0	1
F31	1	0	0	0	1	0	0	0	1	1	1	1	1	0	0	0
F32	0	1	0	0	0	1	0	0	1	1	1	1	0	1	0	0
F33	0	0	1	0	0	0	1	0	1	1	1	1	0	0	1	0
F34	0	0	0	1	0	0	0	1	1	1	1	1	0	0	0	1
F41	1	0	0	0	1	0	0	0	1	0	0	0	1	1	1	1
F42	0	1	0	0	0	1	0	0	0	1	0	0	1	1	1	1
F43	0	0	1	0	0	0	1	0	0	0	1	0	1	1	1	1
F44	0	0	0	1	0	0	0	1	0	0	0	1	1	1	1	1

FIG. 7

FPGA INTERCONNECTION

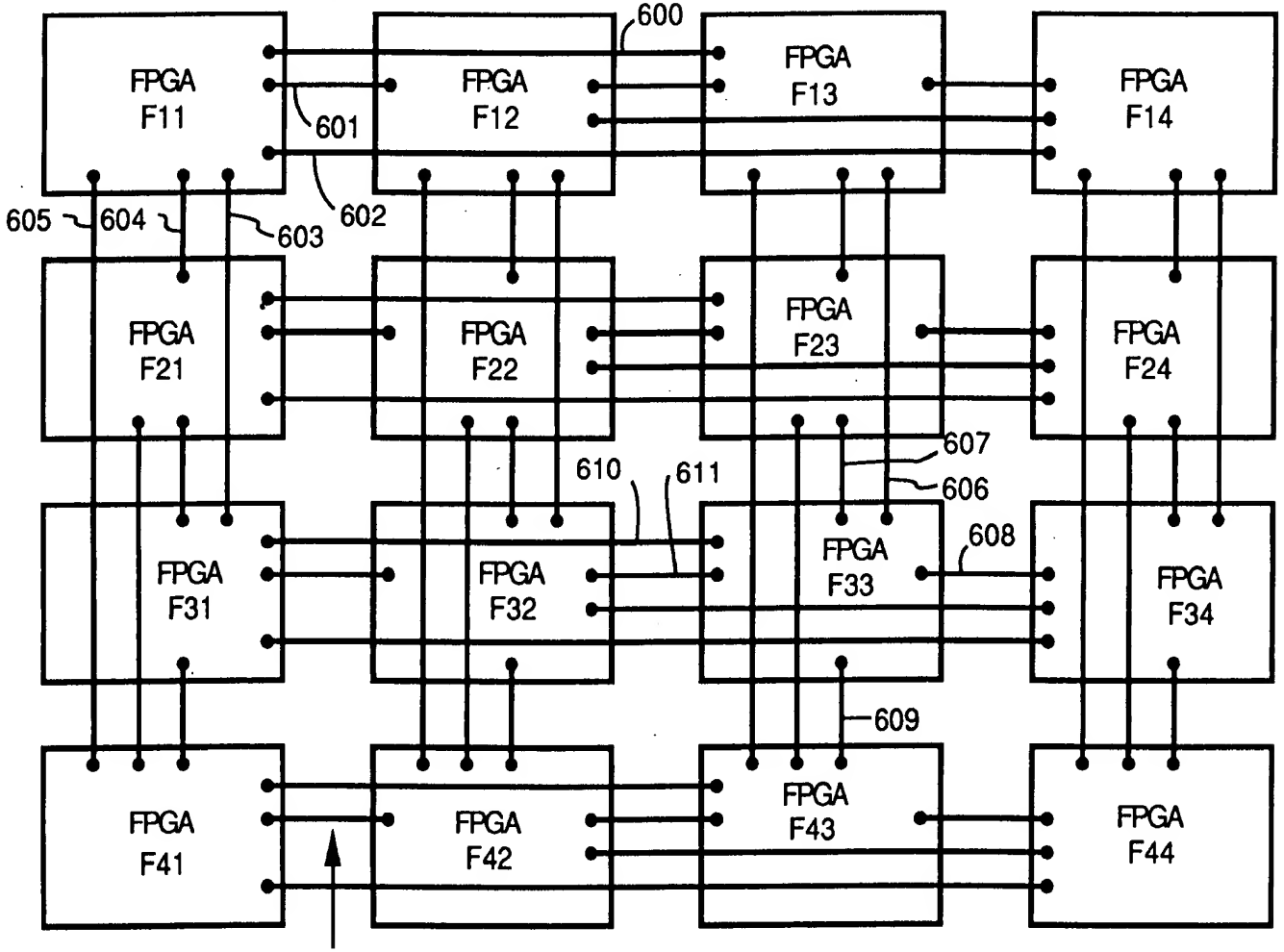


FIG. 8

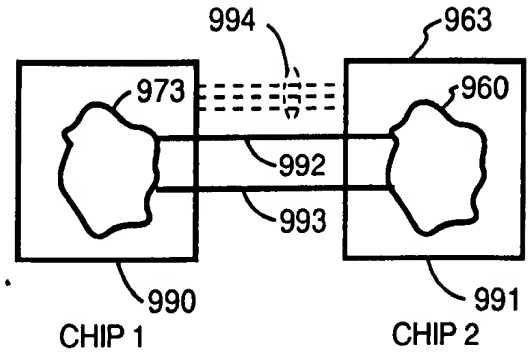

994
963
973
960
992
993
990
991
CHIP 1
CHIP 2

FIG. 9(a)

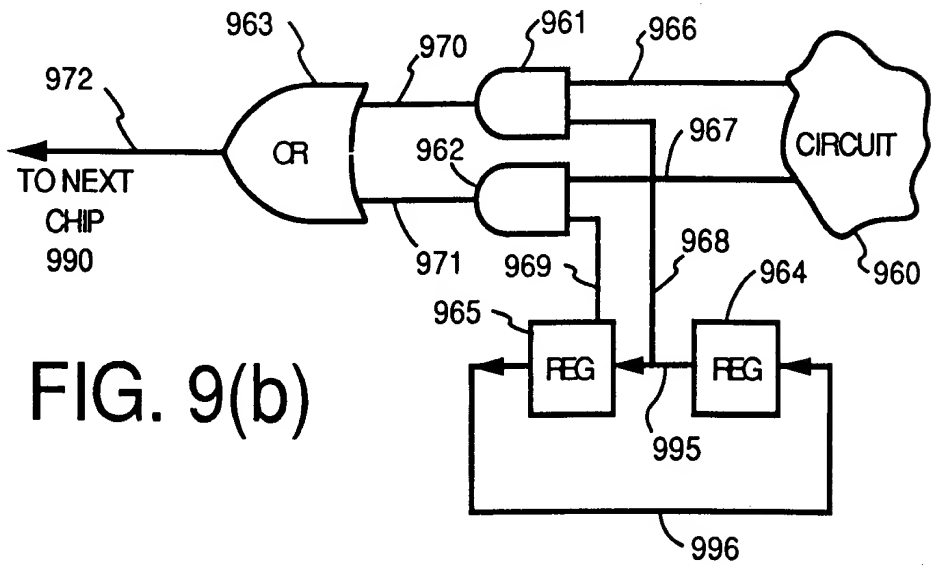

963
970
961
966
972
OR
962
967
CIRCUIT
TO NEXT
CHIP
990
971
969
968
964
960
965
REG
REG
995
996

FIG. 9(b)

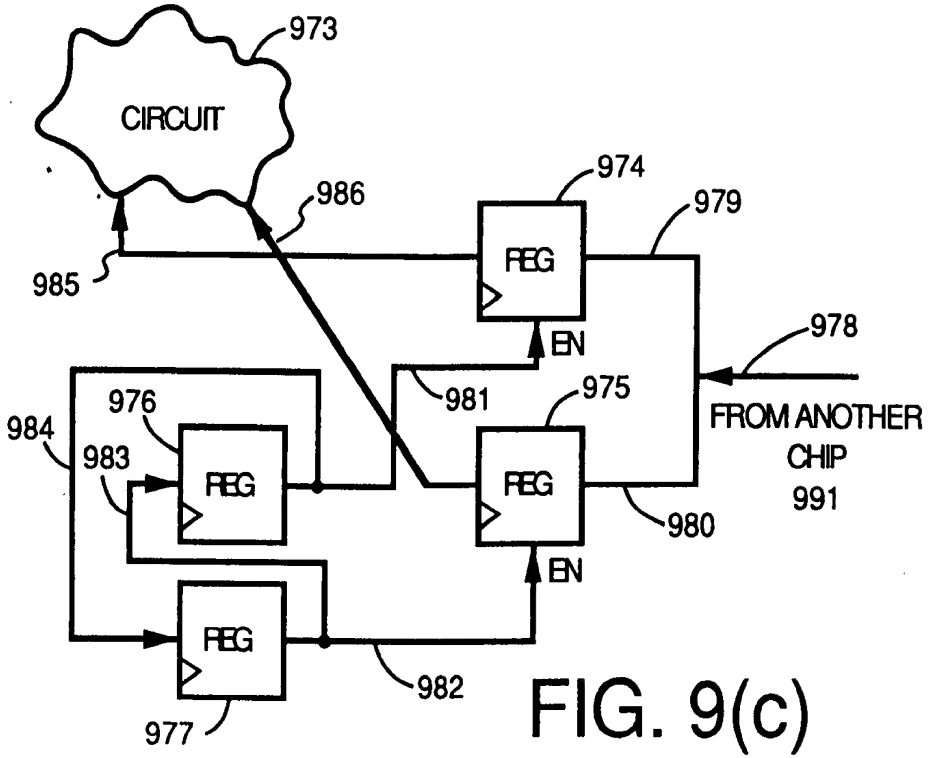


FIG. 9(c)

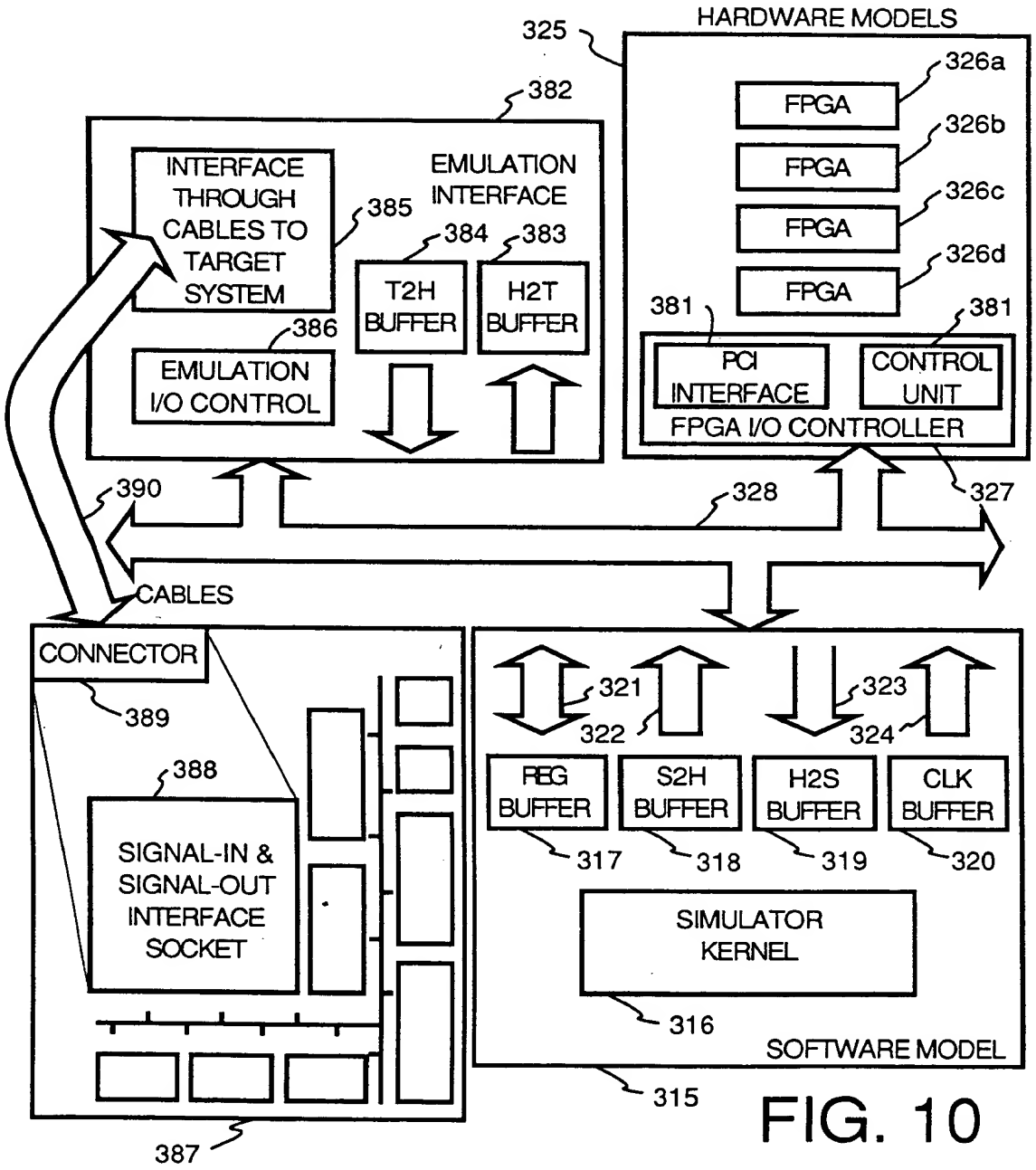

HARDWARE MODELS
325
326a
FPGA
326b
FPGA
326c
FPGA
326d
FPGA
382
INTERFACE THROUGH CABLES TO TARGET SYSTEM
385
EMULATION INTERFACE
384
383
T2H BUFFER
H2T BUFFER
386
EMULATION I/O CONTROL
381
381
PCI INTERFACE
CONTROL UNIT
FPGA I/O CONTROLLER
390
328
327
CABLES
CONNECTOR
389
388
SIGNAL-IN & SIGNAL-OUT INTERFACE SOCKET
321
322
323
324
REG BUFFER
S2H BUFFER
H2S BUFFER
CLK BUFFER
317
318
319
320
SIMULATOR KERNEL
316
SOFTWARE MODEL
315
387

FIG. 10

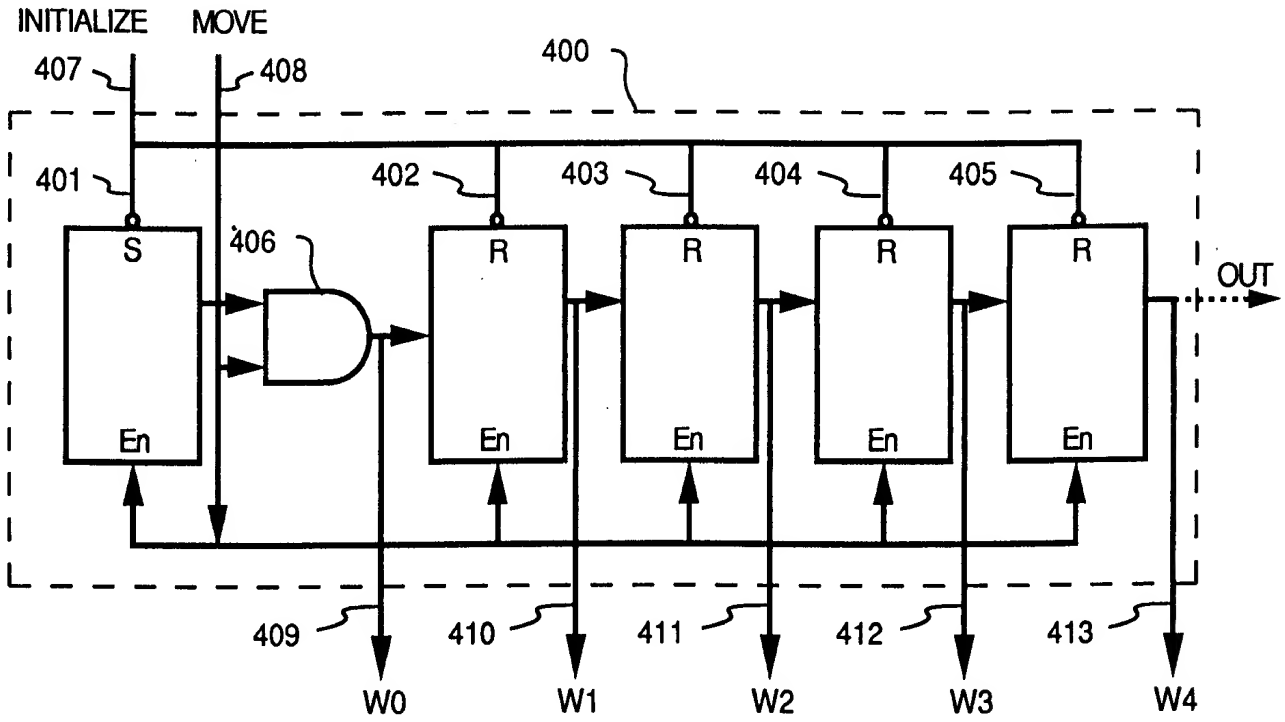


FIG. 11

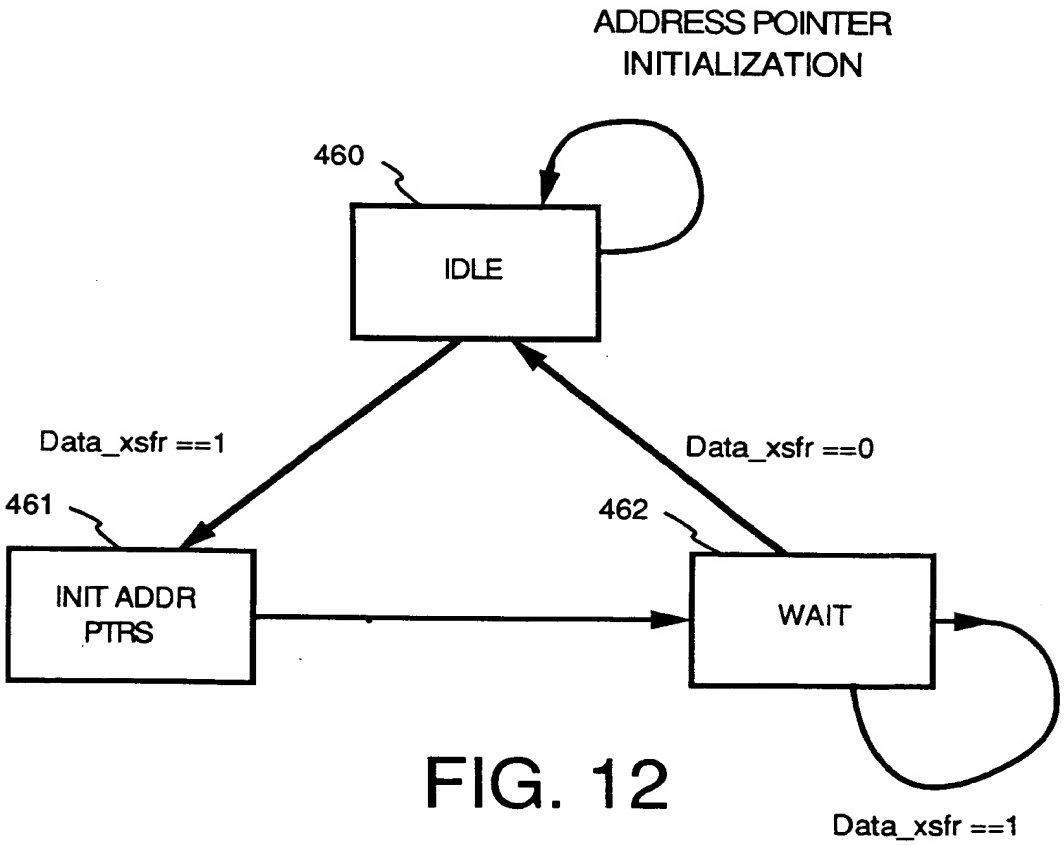


FIG. 12

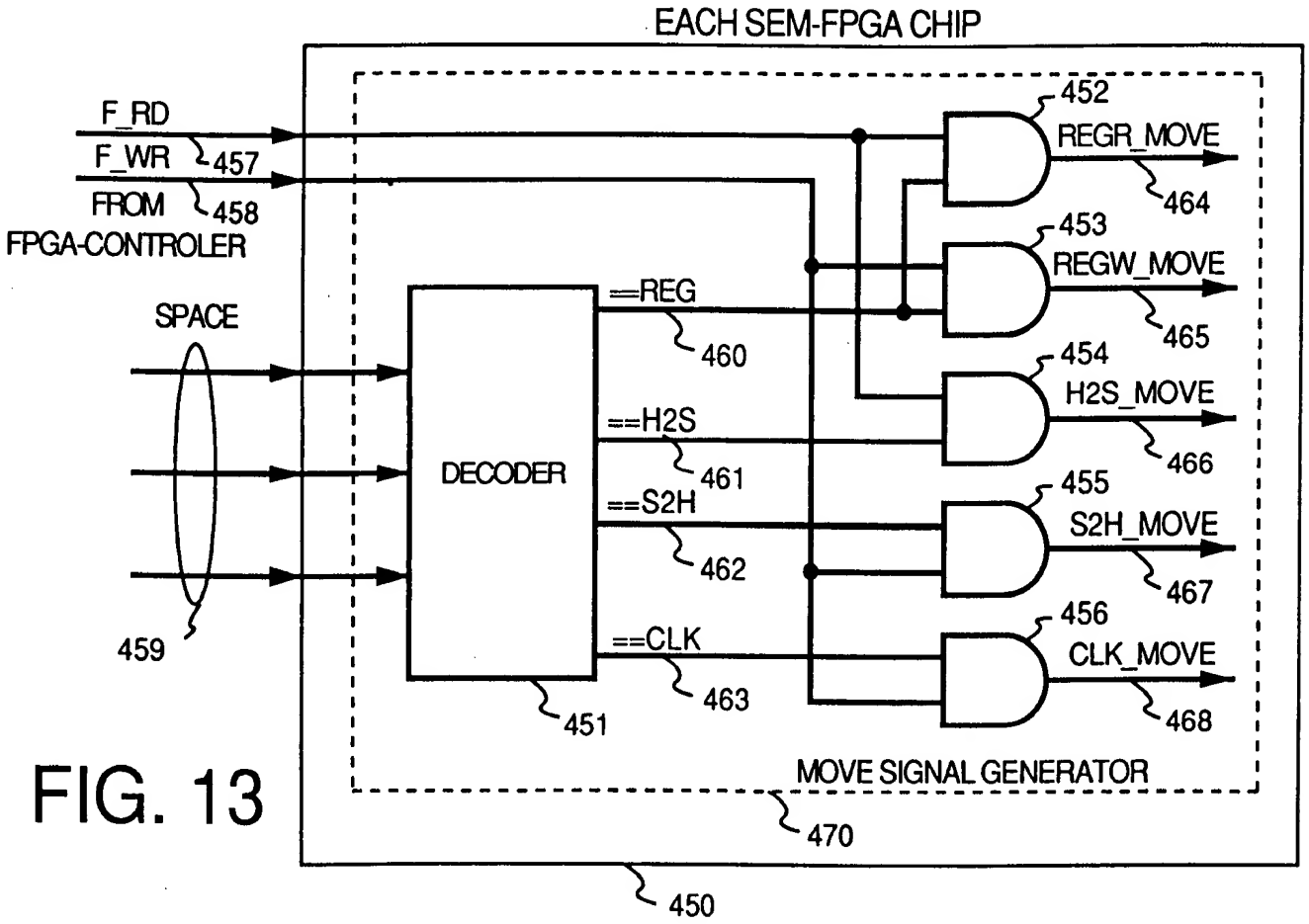

EACH SEM-FPGA CHIP
F_RD
F_WR
457
FROM
458
FPGA-CONTROLER
SPACE
459
DECODER
451
==REG
460
==H2S
461
==S2H
462
==CLK
463
452
REGR_MOVE
464
453
REGW_MOVE
465
454
H2S_MOVE
466
455
S2H_MOVE
467
456
CLK_MOVE
468
MOVE SIGNAL GENERATOR
470
450

FIG. 13

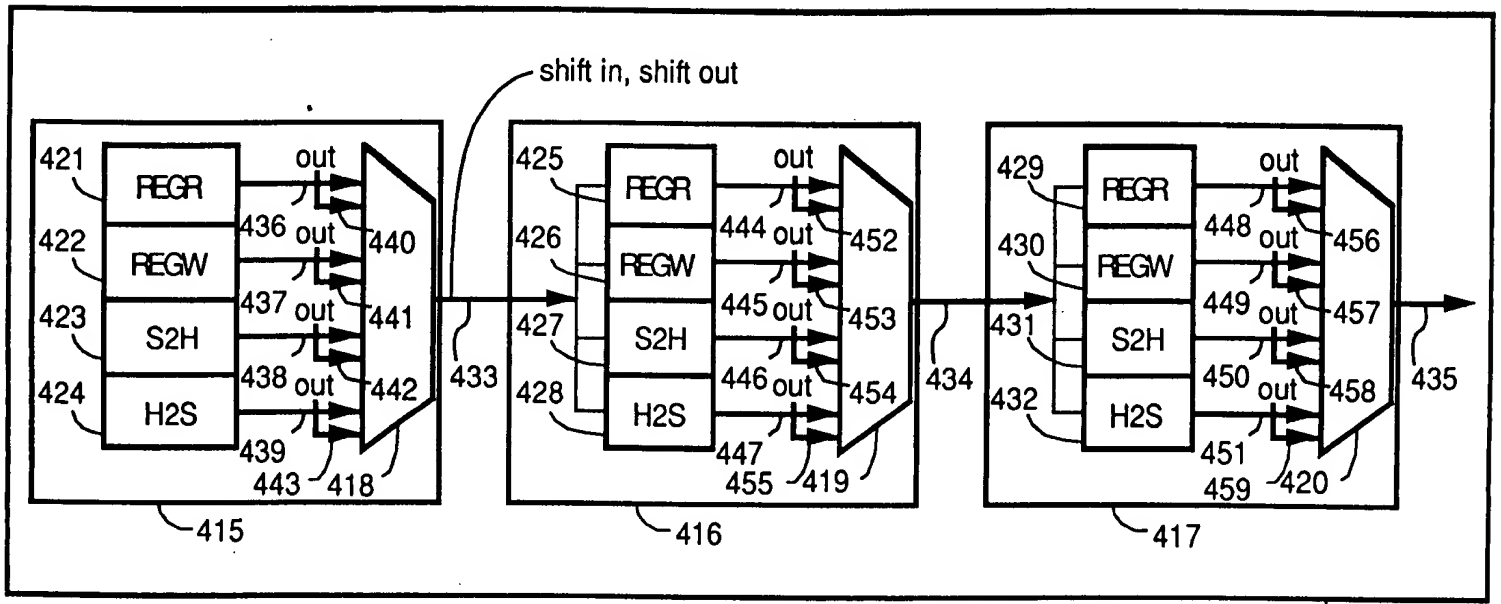


FIG. 14

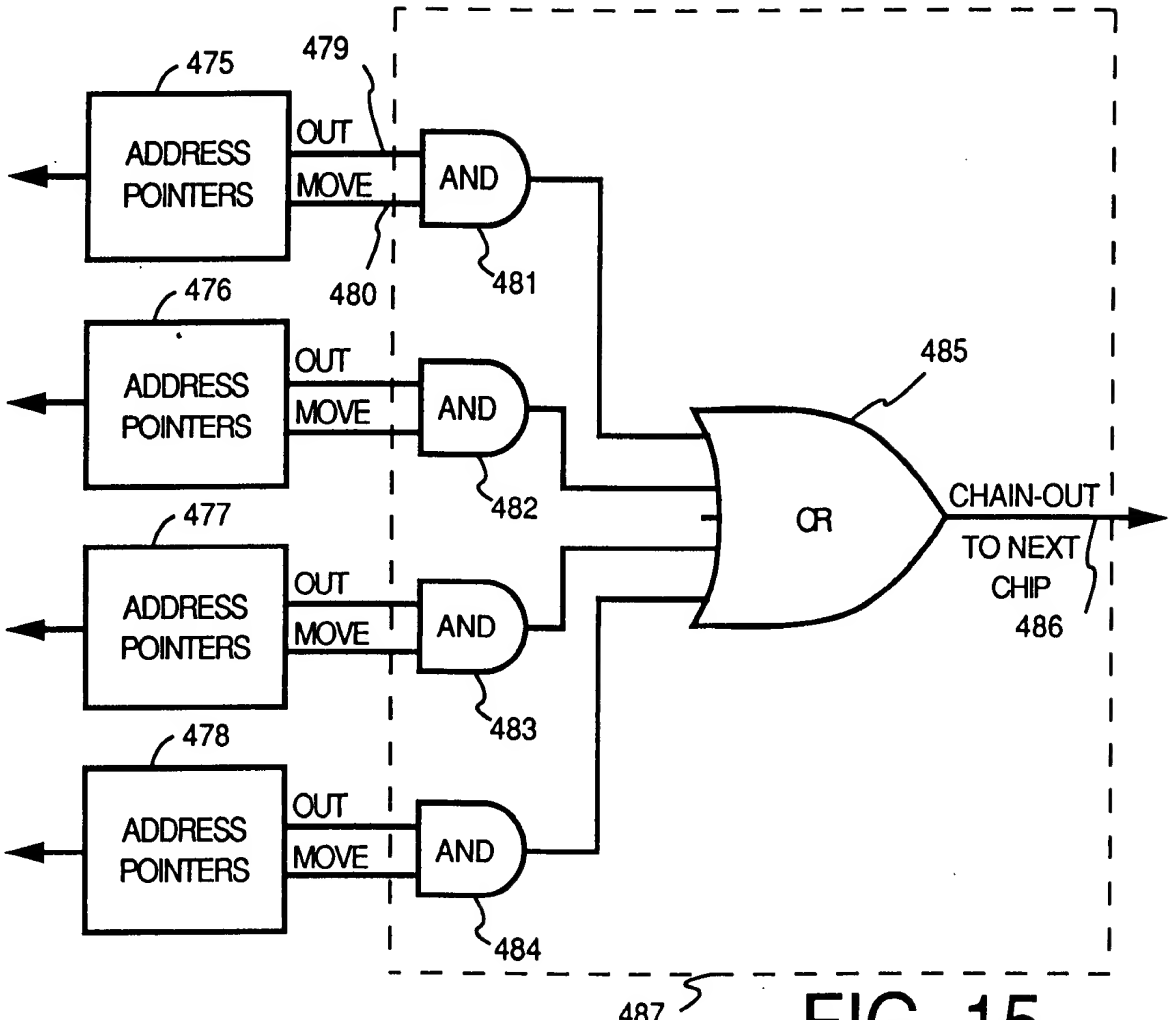


FIG. 15

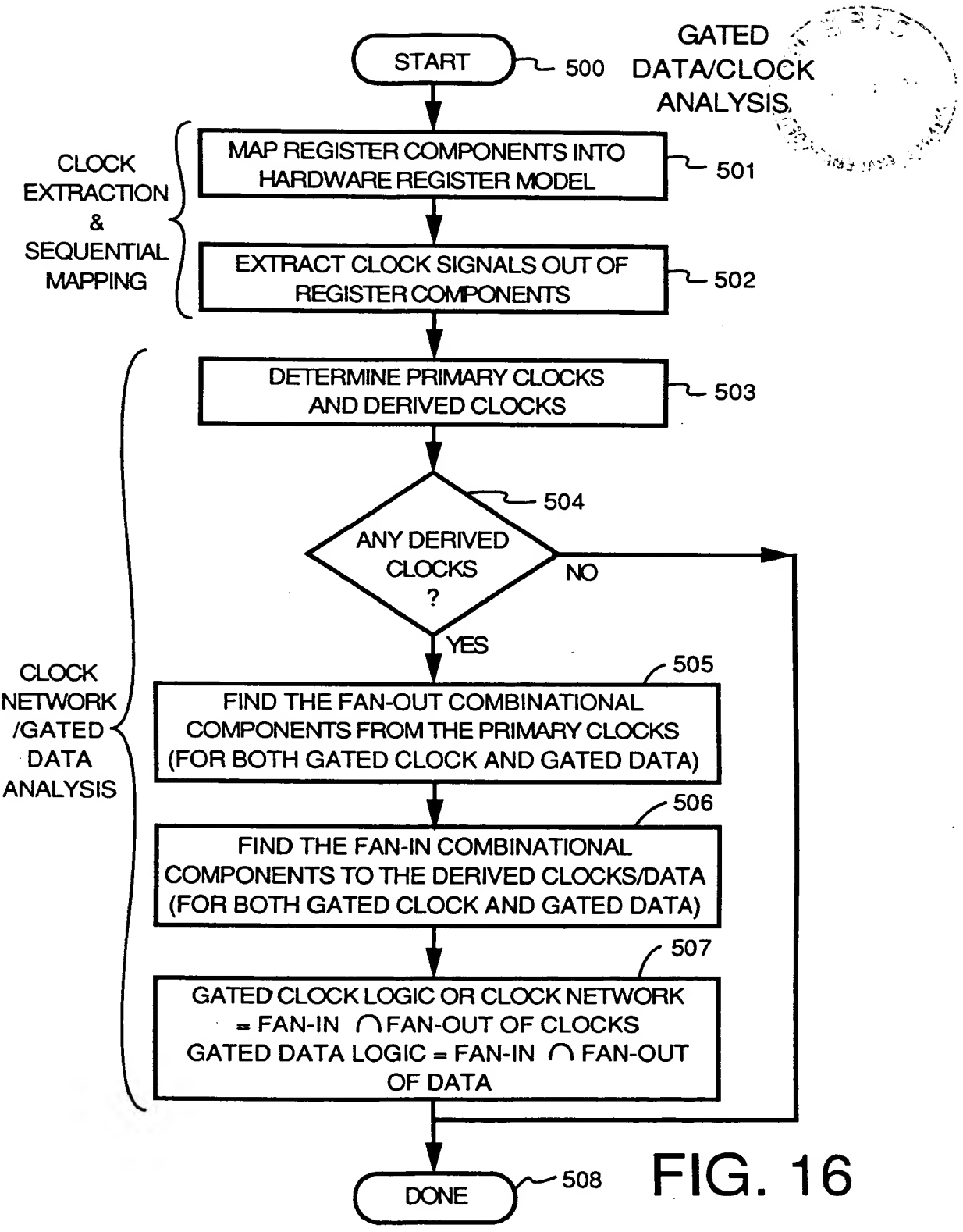

GATED DATA/CLOCK ANALYSIS
START
500
CLOCK EXTRACTION & SEQUENTIAL MAPPING
MAP REGISTER COMPONENTS INTO HARDWARE REGISTER MODEL
501
EXTRACT CLOCK SIGNALS OUT OF REGISTER COMPONENTS
502
DETERMINE PRIMARY CLOCKS AND DERIVED CLOCKS
503
504
ANY DERIVED CLOCKS ?
NO
YES
CLOCK NETWORK /GATED DATA ANALYSIS
505
FIND THE FAN-OUT COMBINATIONAL COMPONENTS FROM THE PRIMARY CLOCKS (FOR BOTH GATED CLOCK AND GATED DATA)
506
FIND THE FAN-IN COMBINATIONAL COMPONENTS TO THE DERIVED CLOCKS/DATA (FOR BOTH GATED CLOCK AND GATED DATA)
507
GATED CLOCK LOGIC OR CLOCK NETWORK = FAN-IN ∩ FAN-OUT OF CLOCKS
GATED DATA LOGIC = FAN-IN ∩ FAN-OUT OF DATA
DONE
508

FIG. 16

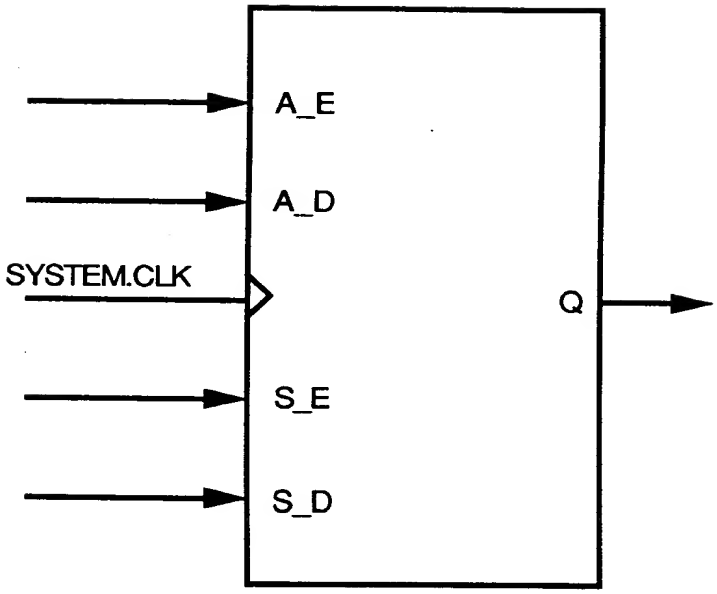


FIG. 17

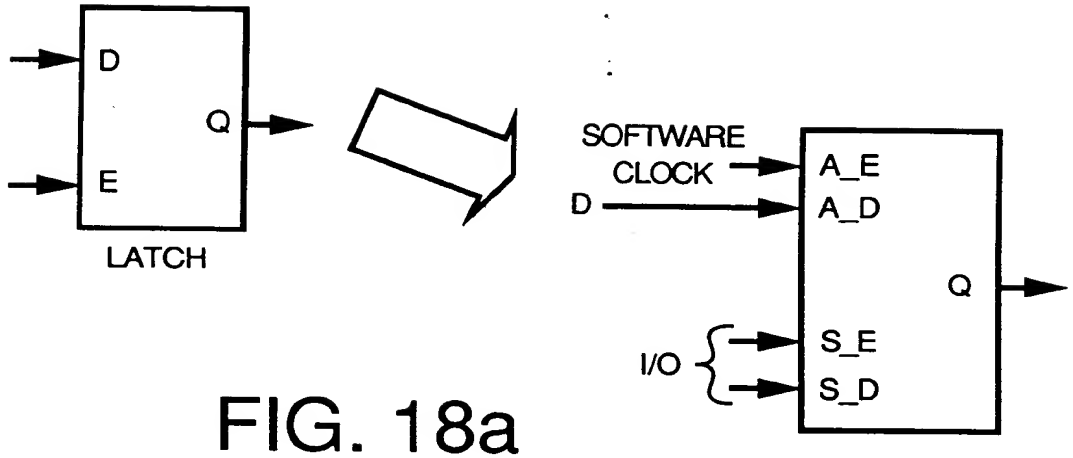


FIG. 18a

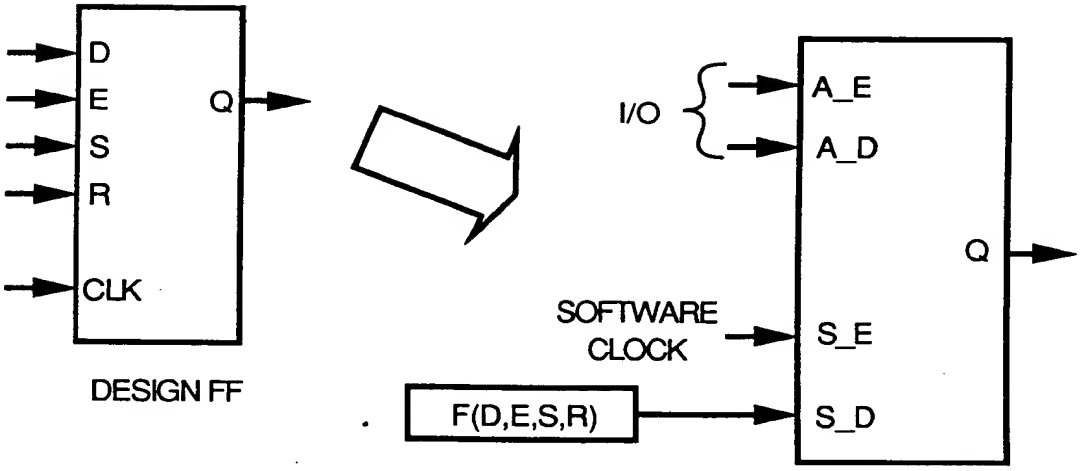


FIG. 18b

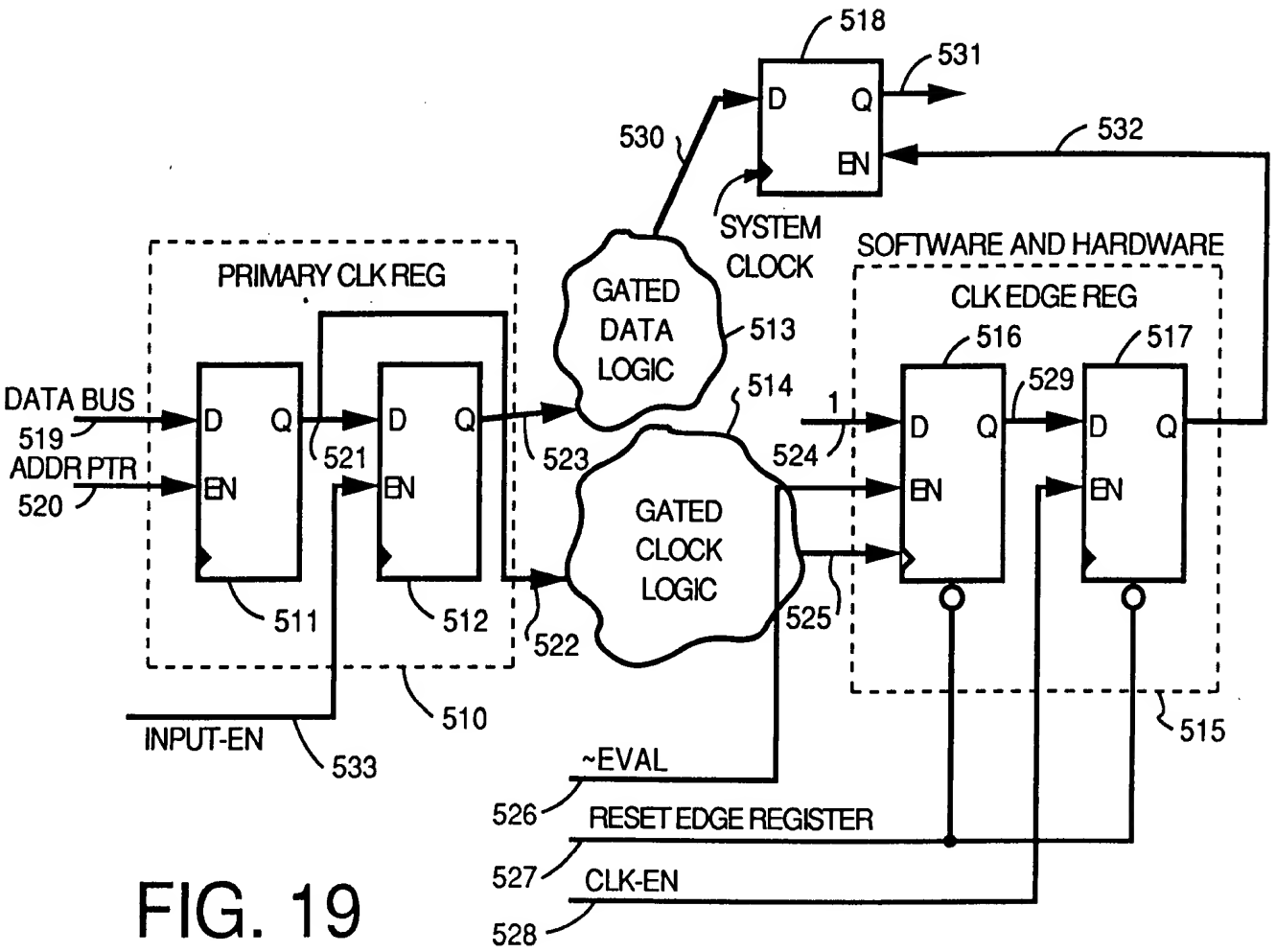


FIG. 19

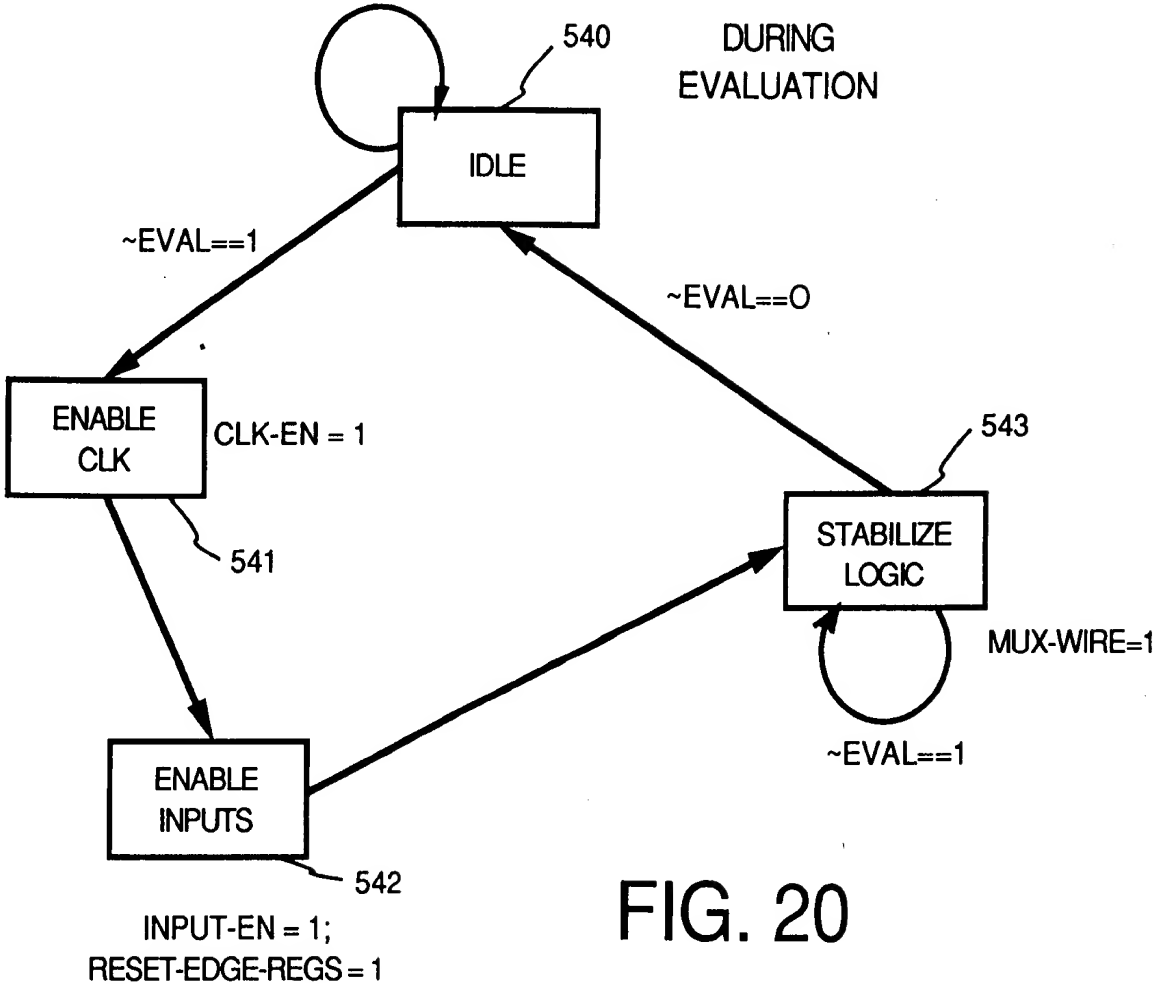

540
DURING
EVALUATION
IDLE
~EVAL==1
~EVAL==0
ENABLE
CLK
CLK-EN = 1
541
543
STABILIZE
LOGIC
MUX-WIRE=1
~EVAL==1
ENABLE
INPUTS
542
INPUT-EN = 1;
RESET-EDGE-REGS = 1

FIG. 20

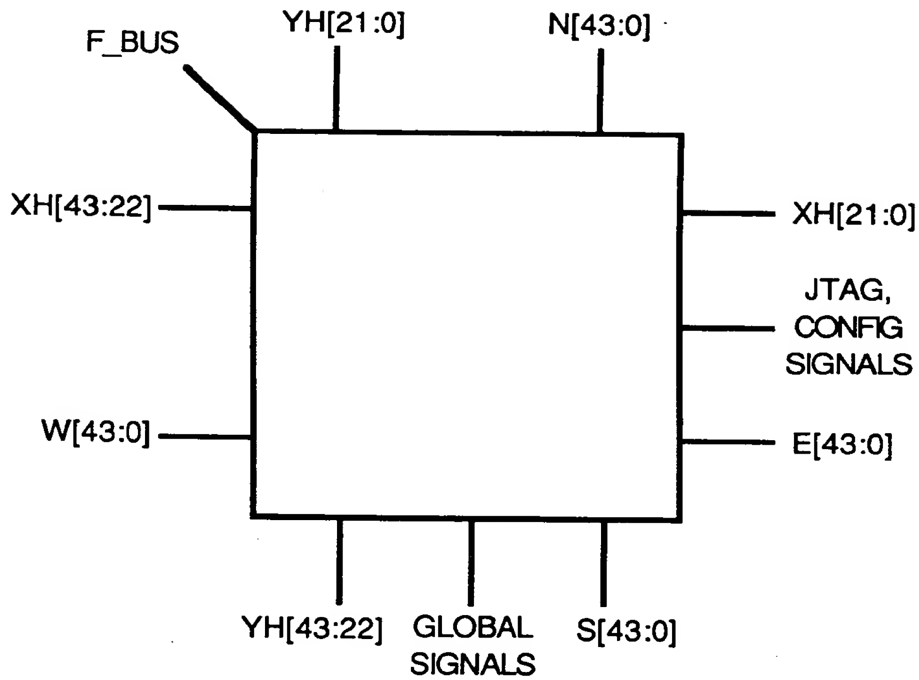

F_BUS
YH[21:0]
N[43:0]
XH[43:22]
XH[21:0]
JTAG, CONFIG SIGNALS
W[43:0]
E[43:0]
YH[43:22]
GLOBAL SIGNALS
S[43:0]

FIG. 21

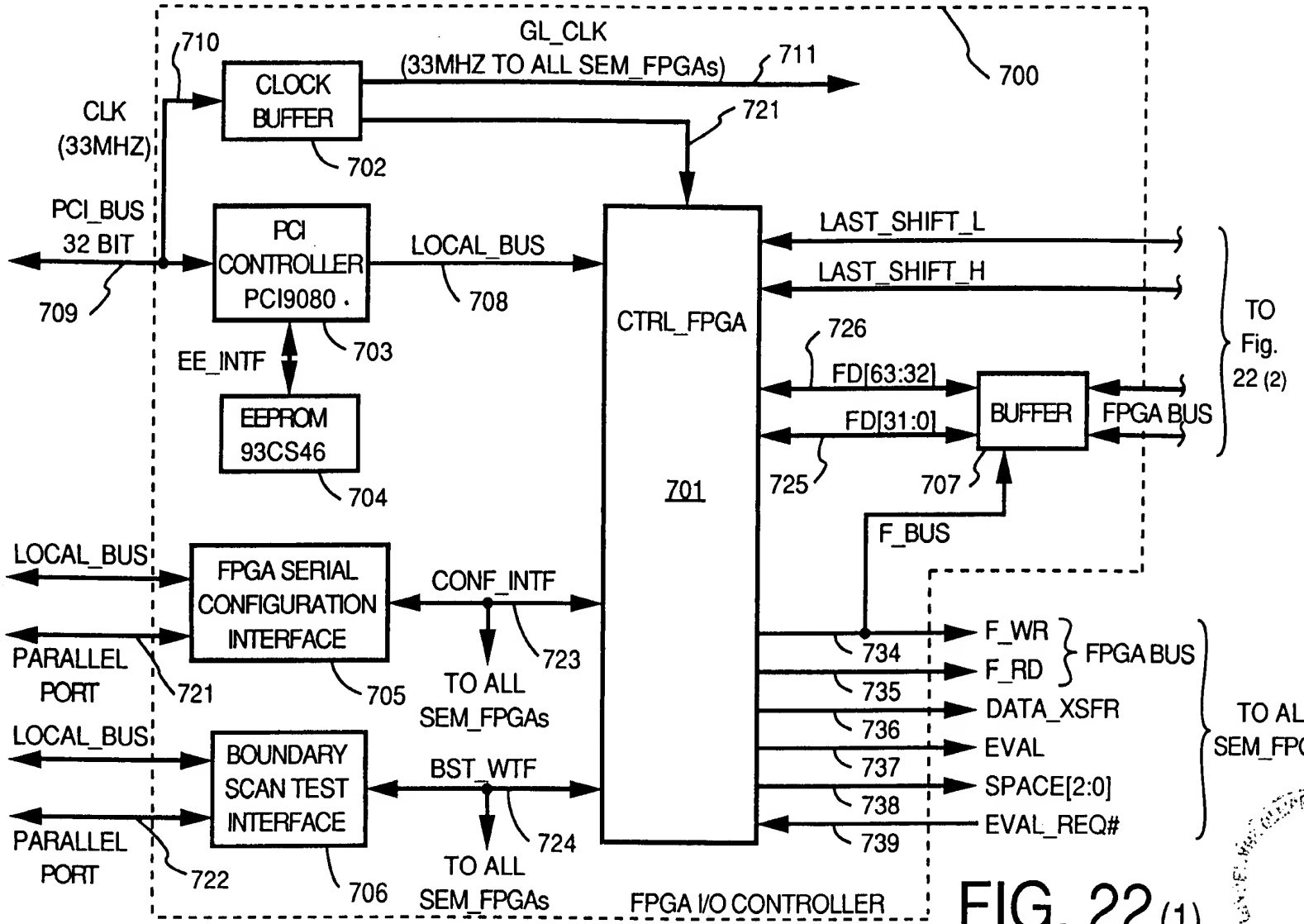

700
710
GL_CLK
(33MHZ TO ALL SEM_FPGAs)
711
CLOCK BUFFER
702
721
CLK (33MHZ)
PCI_BUS 32 BIT
709
PCI CONTROLLER PCI9080
703
LOCAL_BUS
708
CTRL_FPGA
701
LAST_SHIFT_L
LAST_SHIFT_H
TO Fig. 22 (2)
726
FD[63:32]
FD[31:0]
BUFFER
FPGA BUS
725
707
F_BUS
EE_INTF
EEPROM 93CS46
704
LOCAL_BUS
FPGA SERIAL CONFIGURATION INTERFACE
CONF_INTF
PARALLEL PORT
721
705
723
TO ALL SEM_FPGAs
LOCAL_BUS
BOUNDARY SCAN TEST INTERFACE
BST_WTF
PARALLEL PORT
722
706
724
TO ALL SEM_FPGAs
FPGA I/O CONTROLLER
734
735
736
737
738
739
F_WR
F_RD
DATA_XSFR
EVAL
SPACE[2:0]
EVAL_REQ#
FPGA BUS
TO ALL SEM_FPGAs

FIG. 22 (1)

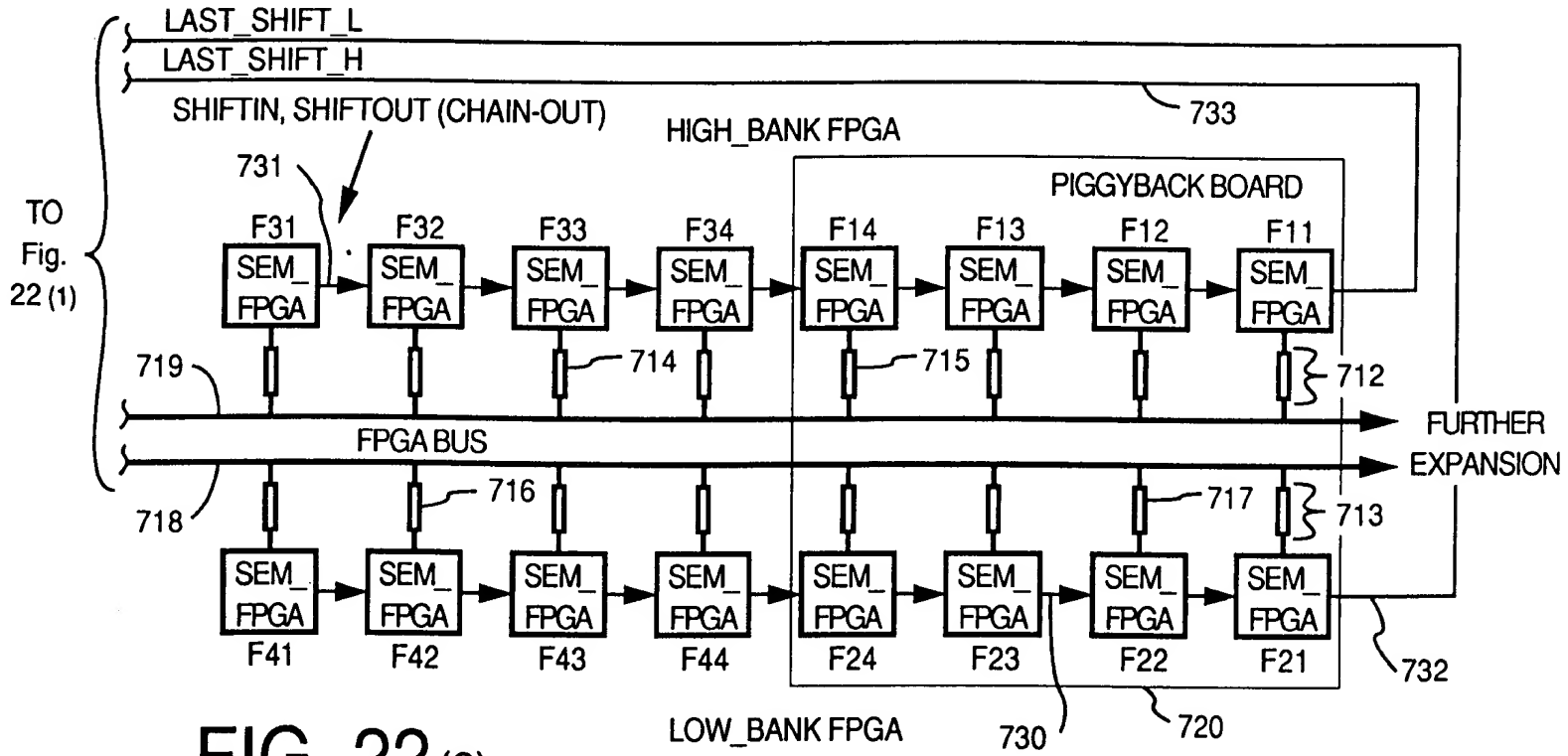

LAST_SHIFT_L
LAST_SHIFT_H
SHIFTIN, SHIFTOUT (CHAIN-OUT)
731
733
HIGH_BANK FPGA
PIGGYBACK BOARD
TO
Fig.
22 (1)
F31
F32
F33
F34
F14
F13
F12
F11
SEM_
FPGA
719
714
715
712
FPGA BUS
FURTHER
EXPANSION
718
716
717
713
F41
F42
F43
F44
F24
F23
F22
F21
732
LOW_BANK FPGA
730
720

FIG. 22 (2)

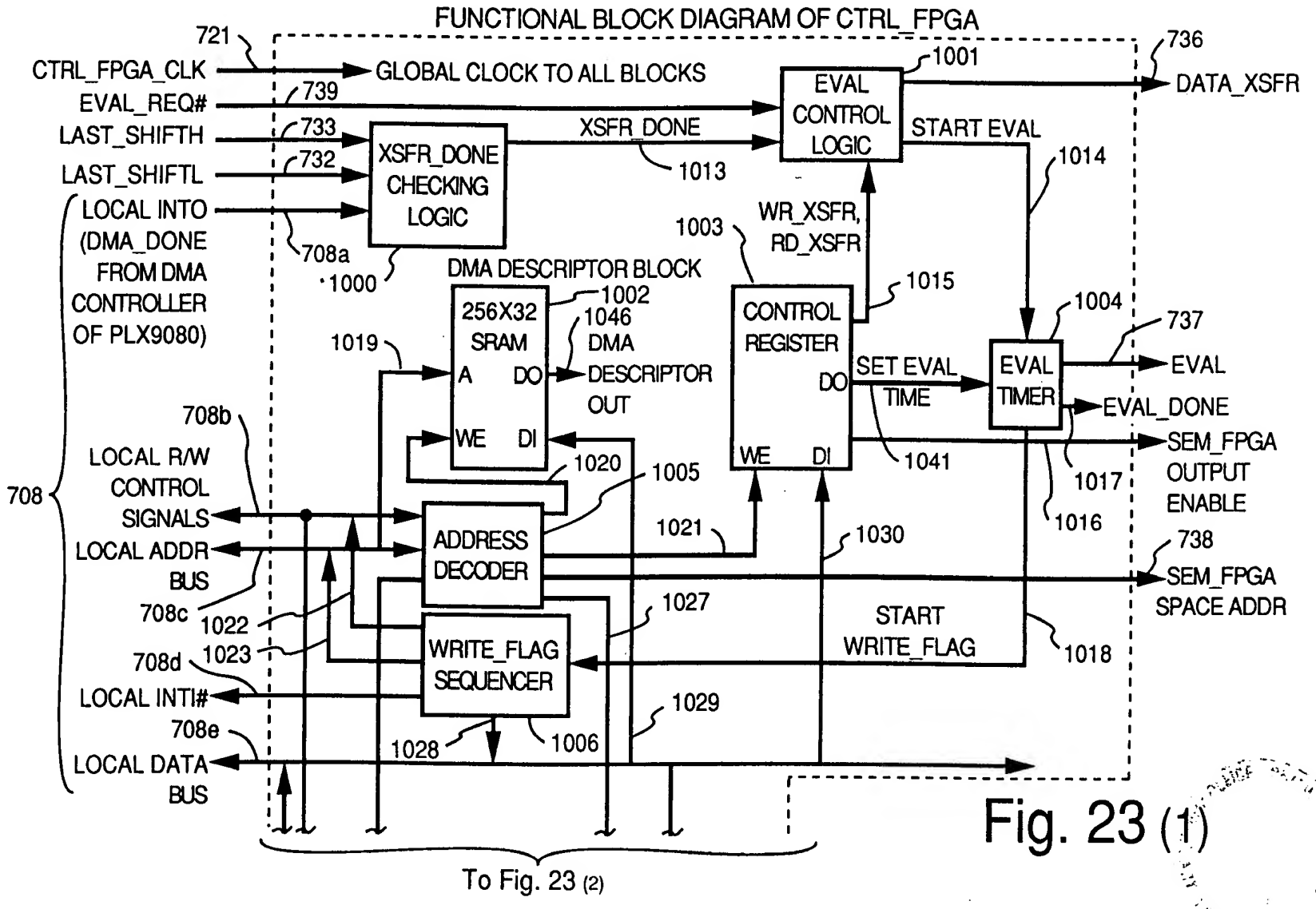

FUNCTIONAL BLOCK DIAGRAM OF CTRL_FPGA
721
CTRL_FPGA_CLK
GLOBAL CLOCK TO ALL BLOCKS
EVAL_REQ#
739
LAST_SHIFTH
733
LAST_SHIFTL
732
LOCAL INTO
(DMA_DONE
FROM DMA
CONTROLLER
OF PLX9080)
708a
XSFR_DONE
CHECKING
LOGIC
1000
XSFR_DONE
1013
EVAL
CONTROL
LOGIC
1001
START EVAL
1014
736
DATA_XSFR
WR_XSFR,
RD_XSFR
1003
1015
DMA DESCRIPTOR BLOCK
256X32
SRAM
A
DO
WE
DI
1002
1046
DMA
DESCRIPTOR
OUT
1019
CONTROL
REGISTER
DO
WE
DI
1004
737
EVAL
EVAL_DONE
SET EVAL
TIME
EVAL
TIMER
1041
1017
1016
SEM_FPGA
OUTPUT
ENABLE
708b
LOCAL R/W
CONTROL
SIGNALS
708
LOCAL ADDR
BUS
708c
1022
1020
1005
ADDRESS
DECODER
1021
1030
738
SEM_FPGA
SPACE ADDR
1027
START
WRITE_FLAG
1018
1023
708d
WRITE_FLAG
SEQUENCER
LOCAL INTI#
1029
708e
1028
1006
LOCAL DATA
BUS
To Fig. 23 (2)

Fig. 23 (1)

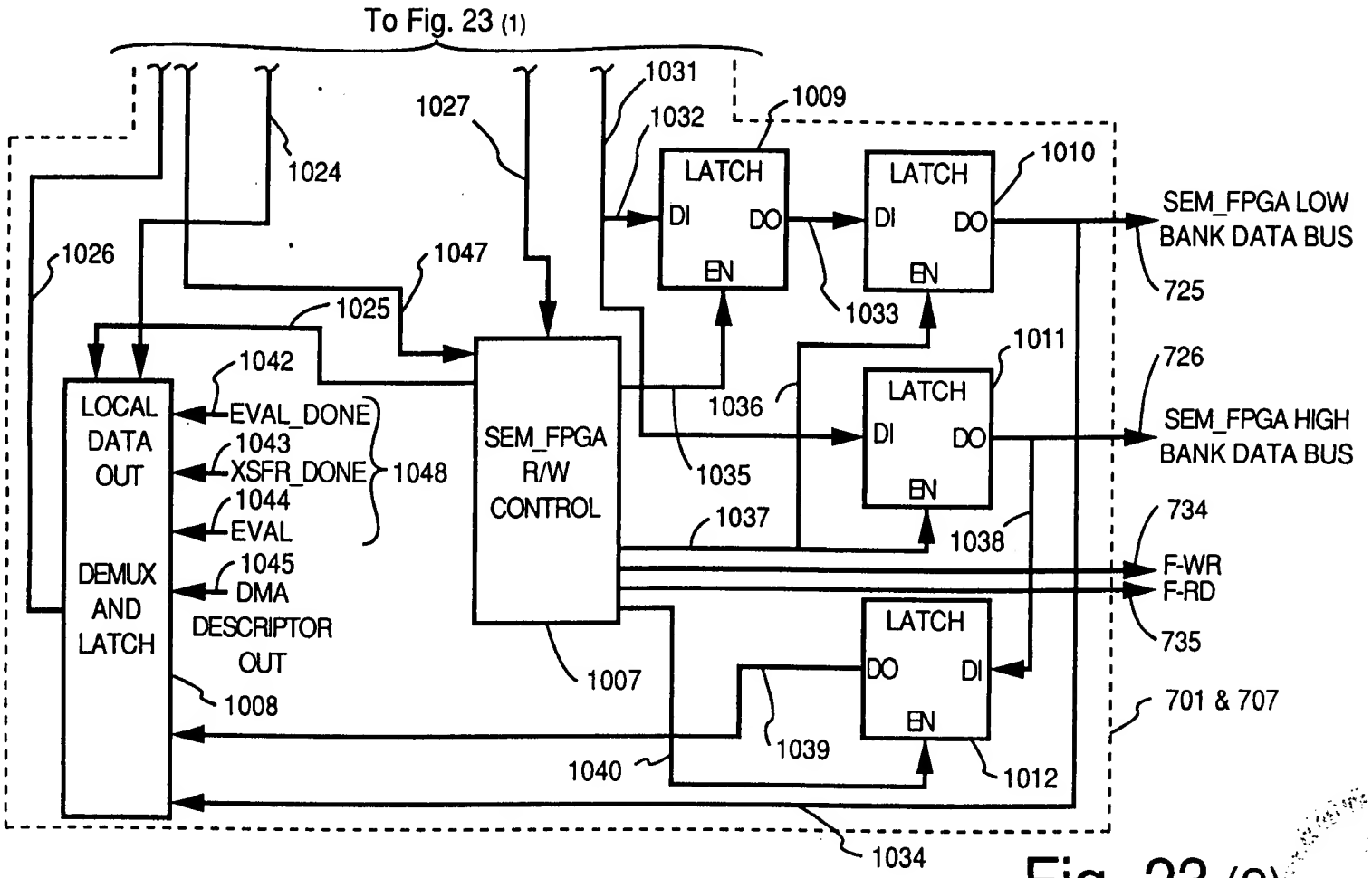

To Fig. 23 (1)
1024
1027
1031
1032
1009
1010
LATCH
DI
DO
EN
LATCH
DI
DO
EN
SEM_FPGA LOW
BANK DATA BUS
725
1026
1047
1025
1033
1011
1042
LOCAL
DATA
OUT
EVAL_DONE
1043
XSFR_DONE
1048
1044
EVAL
1045
DMA
DESCRIPTOR
OUT
DEMUX
AND
LATCH
1008
SEM_FPGA
R/W
CONTROL
1036
1035
1037
LATCH
DI
DO
EN
726
SEM_FPGA HIGH
BANK DATA BUS
1038
734
F-WR
F-RD
735
1007
LATCH
DO
DI
EN
701 & 707
1039
1040
1012
1034

Fig. 23 (2)

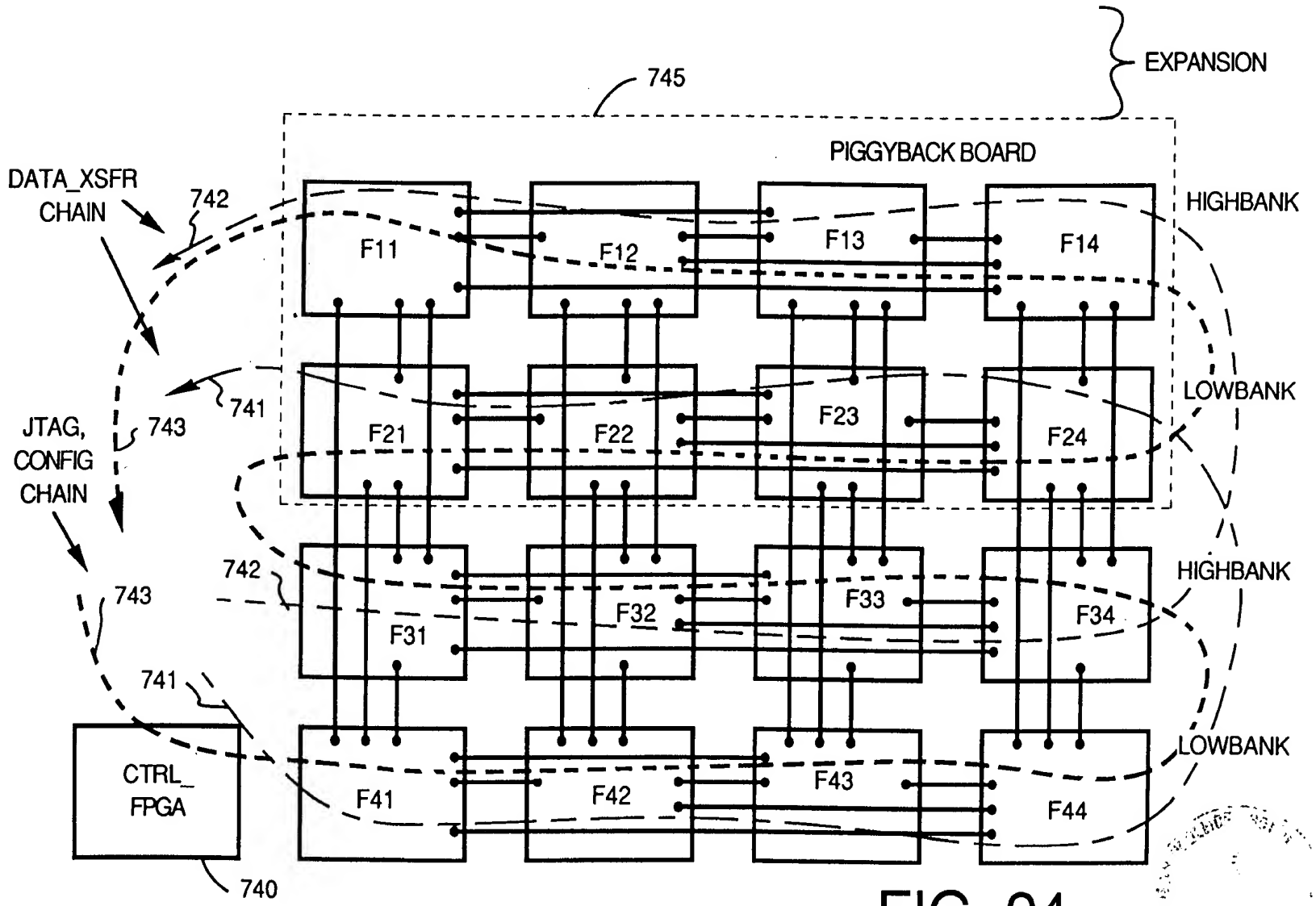

EXPANSION
745
PIGGYBACK BOARD
DATA_XSFR
CHAIN
742
HIGHBANK
F11
F12
F13
F14
741
LOWBANK
JTAG,
CONFIG
CHAIN
743
F21
F22
F23
F24
742
HIGHBANK
743
F31
F32
F33
F34
741
LOWBANK
CTRL_
FPGA
F41
F42
F43
F44
740

FIG. 24

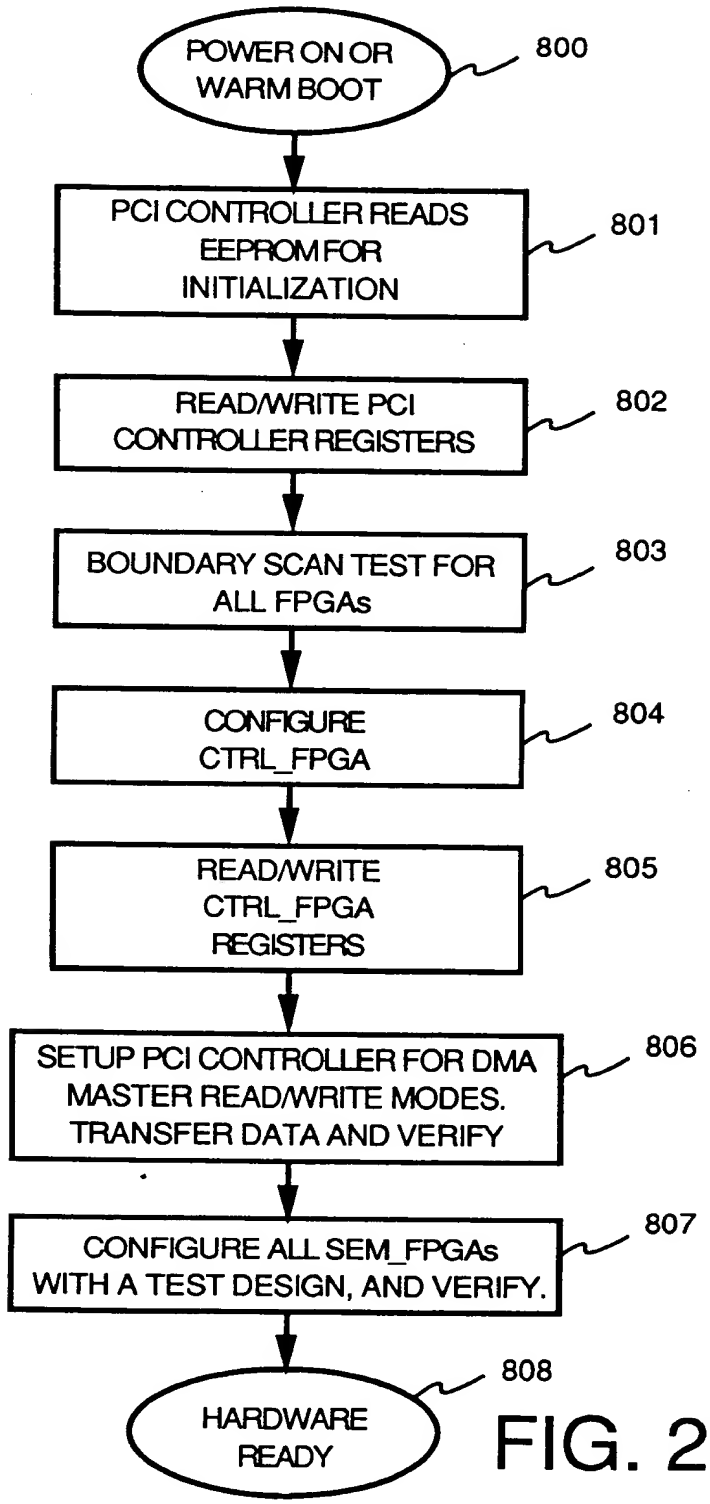

HARDWARE START-UP
POWER ON OR WARM BOOT
800
PCI CONTROLLER READS EEPROM FOR INITIALIZATION
801
READ/WRITE PCI CONTROLLER REGISTERS
802
BOUNDARY SCAN TEST FOR ALL FPGAs
803
CONFIGURE CTRL_FPGA
804
READ/WRITE CTRL_FPGA REGISTERS
805
SETUP PCI CONTROLLER FOR DMA MASTER READ/WRITE MODES. TRANSFER DATA AND VERIFY
806
CONFIGURE ALL SEM_FPGAs WITH A TEST DESIGN, AND VERIFY.
807
HARDWARE READY
808

FIG. 25

```
module register (clock, reset, d, q);
input clock, d, reset;
output q;
reg q;

always@(posedge clock or negedge reset)
   if(~reset)
      q = 0;
   else
      q = d;
endmodule

module example;
   wire d1, d2, d3;
   wire q1, q2, q3;
   reg sigin;
   wire sigout;
   reg clk, reset;

   register reg1 (clk, reset, d1, q1);
   register reg2 (clk, reset, d2, q2);
   register reg3 (clk, reset, d3, q3);

   assign d1 = sigin ^ q3;
   assign d2 = q1 ^ q3;
   assign d3 = q2 ^ q3;
   assign sigout = q3;

   // a clock generator
   always
   begin
      clk = 0;
      #5;
      clk = 1;
      #5;
   end

   // a signal generator
   always
   begin
      #10;
      sigin = $random;
   end

   // initialization
   initial
   begin
      reset = 0;
      sigin = 0;
      #1;
      reset =1;
      #5;
      $monitor($time, " %b, %b", sigin, sigout);
      #1000 $finish;
   end
   end module
```

Fig. 26

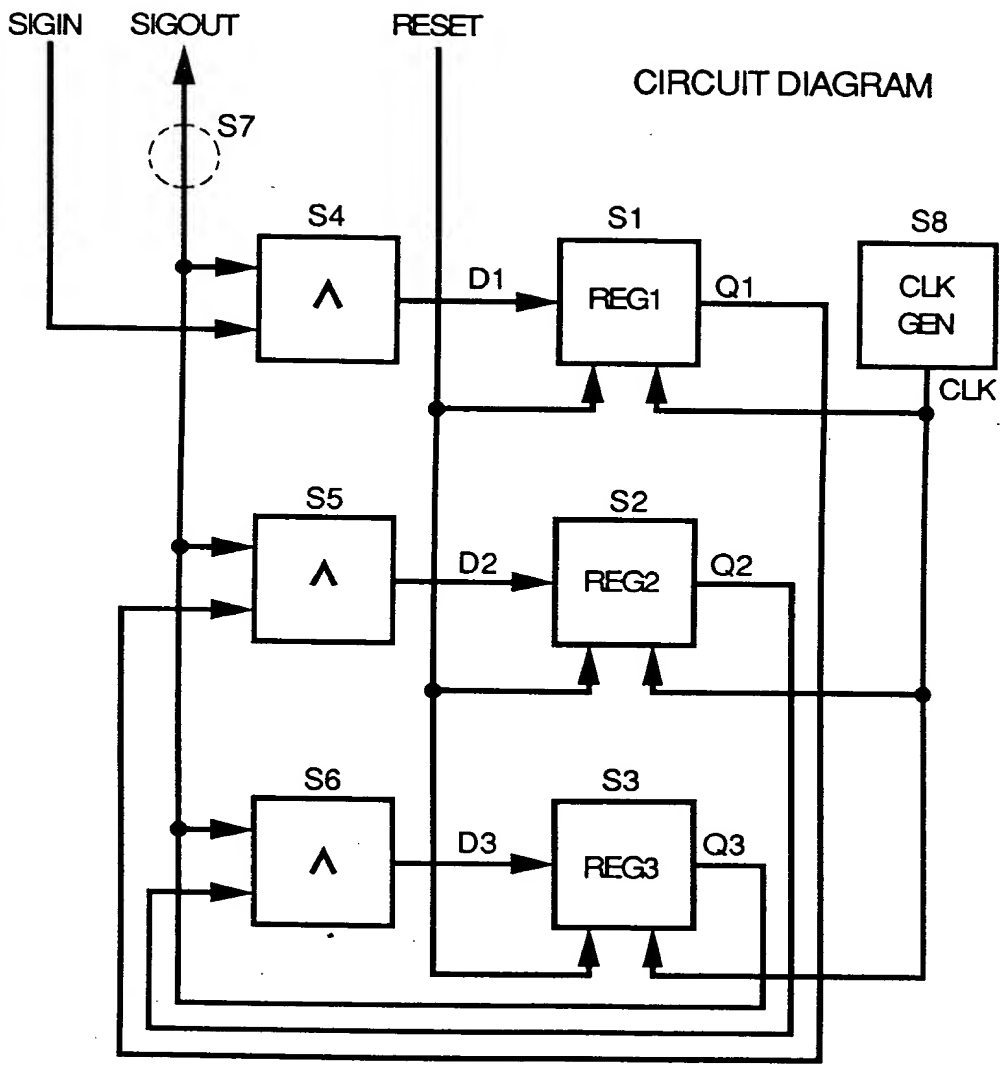

SIGIN
SIGOUT
RESET
CIRCUIT DIAGRAM
S7
S4
S1
S8
D1
REG1
Q1
CLK
GEN
CLK
S5
S2
D2
REG2
Q2
S6
S3
D3
REG3
Q3

FIG. 27

```
module register (clock, reset, d, q);
input clock, d, reset;
output q;
reg q;

always@(post edge clock or negedge reset)
   if(~reset)
      q = 0
   else
      q = d;

endmodule
```

Register definition — 900

```
module example;
   wire d1, d2, d3;
   ware q1, q2, q3;
```

Wire interconnection info — 907

```
   reg sigin;
   wire sigout;
   reg clk, reset;
```

Test-bench input -- 908 (reg sigin)

Test-bench output -- 909 (wire sigout)

```
S1  register reg 1 (clk, reset, d1, q1);
S2  register reg 2 (clk, reset, d2, q2);
S3  register reg 3 (clk, reset, d3, q3);
```

Register component — 901

```
S4  assign d1 = sigin ^ q3;
S5  assign d2 = q1 ^ 3;
S6  assign d3 = q2 ^ q3;
S7  assign signout = q3;
```

Combinational component — 902

```
S8    // a clock generator
      always
      begin
         clk = 0;
         #5;
         clk = 1;
         #5;
      end
```

Clock component — 903

```
S9    // a signal generator
      always
      begin
         #10;
         sigin = $random;
      end
```

Test-bench component (Driver) — 904

```
      // initialization
      initial
S10   begin
         reset = 0;
         sigin = 0;
         #1;
S11      reset = 1;
         #5;
S12      $monitor($time, "%b, %b", sigin, sigout)
         #1000 $finish;
      end
      end module
```

Test-bench component (initialization) — 905

Test-bench component (monitor) — 906

Fig. 28

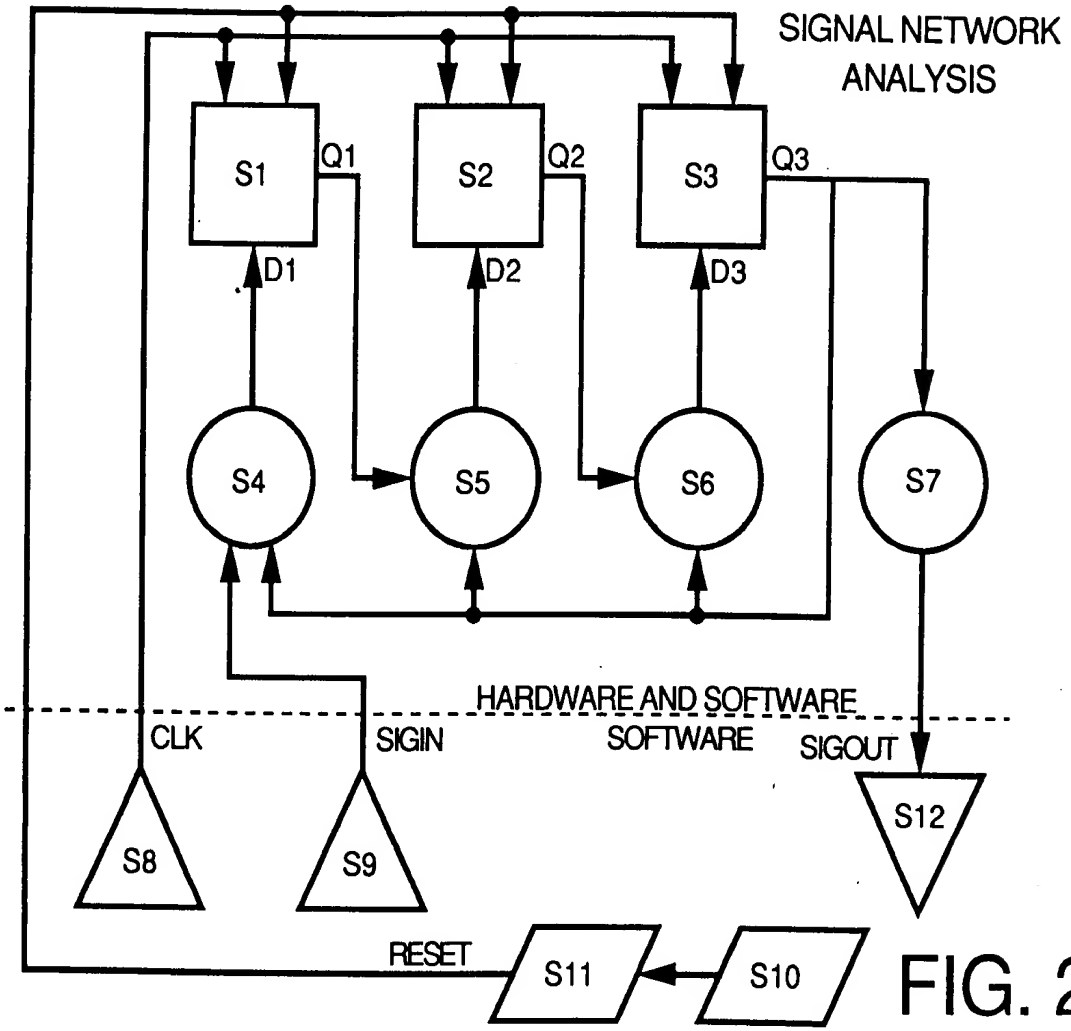


FIG. 29

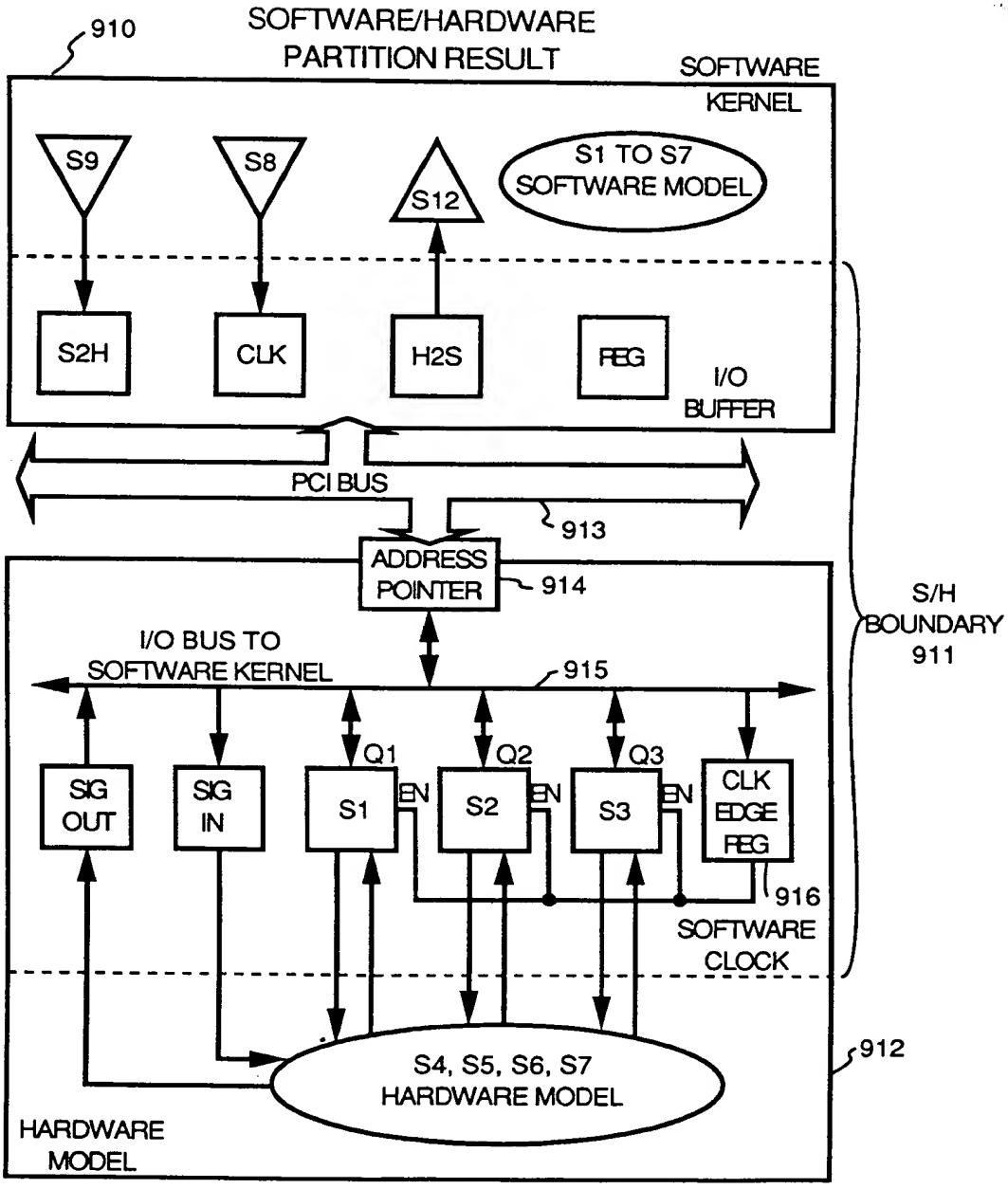


FIG. 30

HARDWARE MODEL

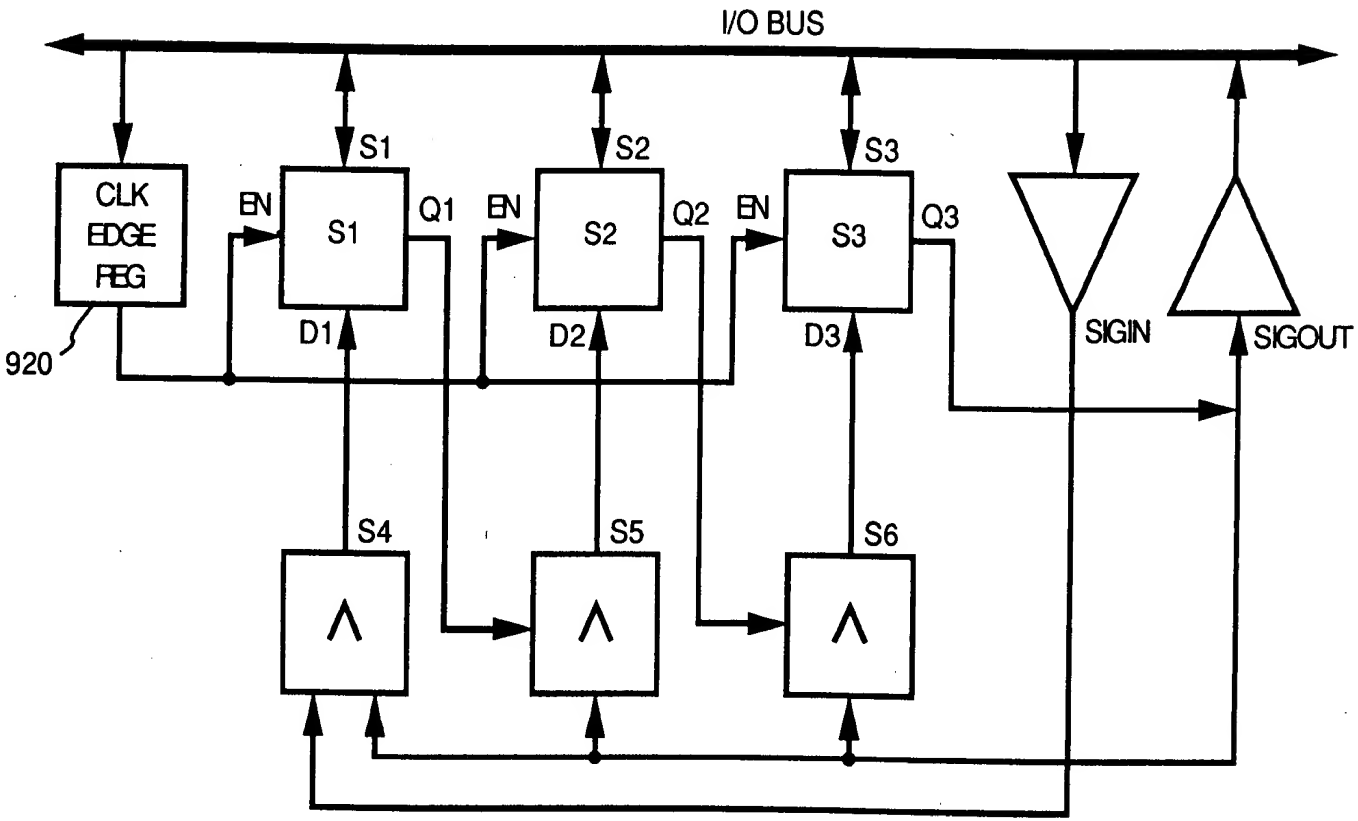
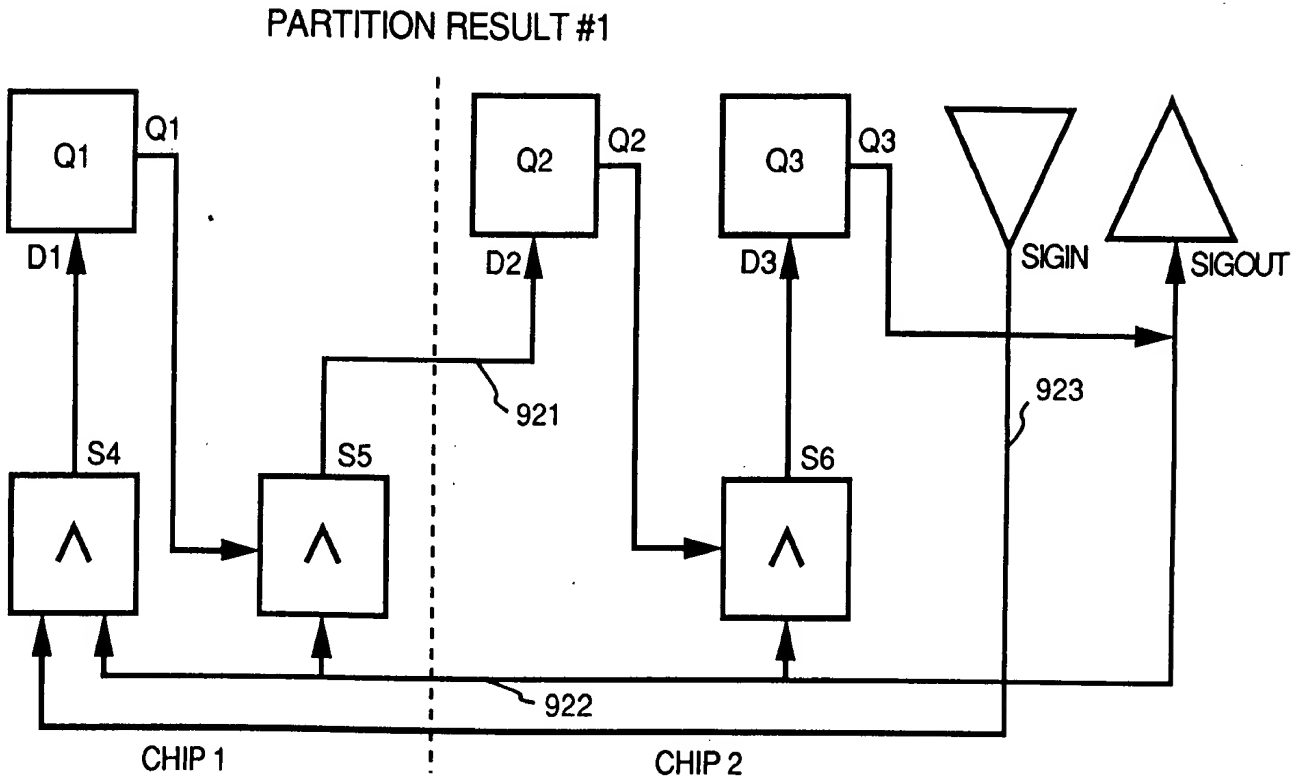


FIG. 31



(IGNORE I/O AND CLOCK EDGE REGISTER)

FIG. 32

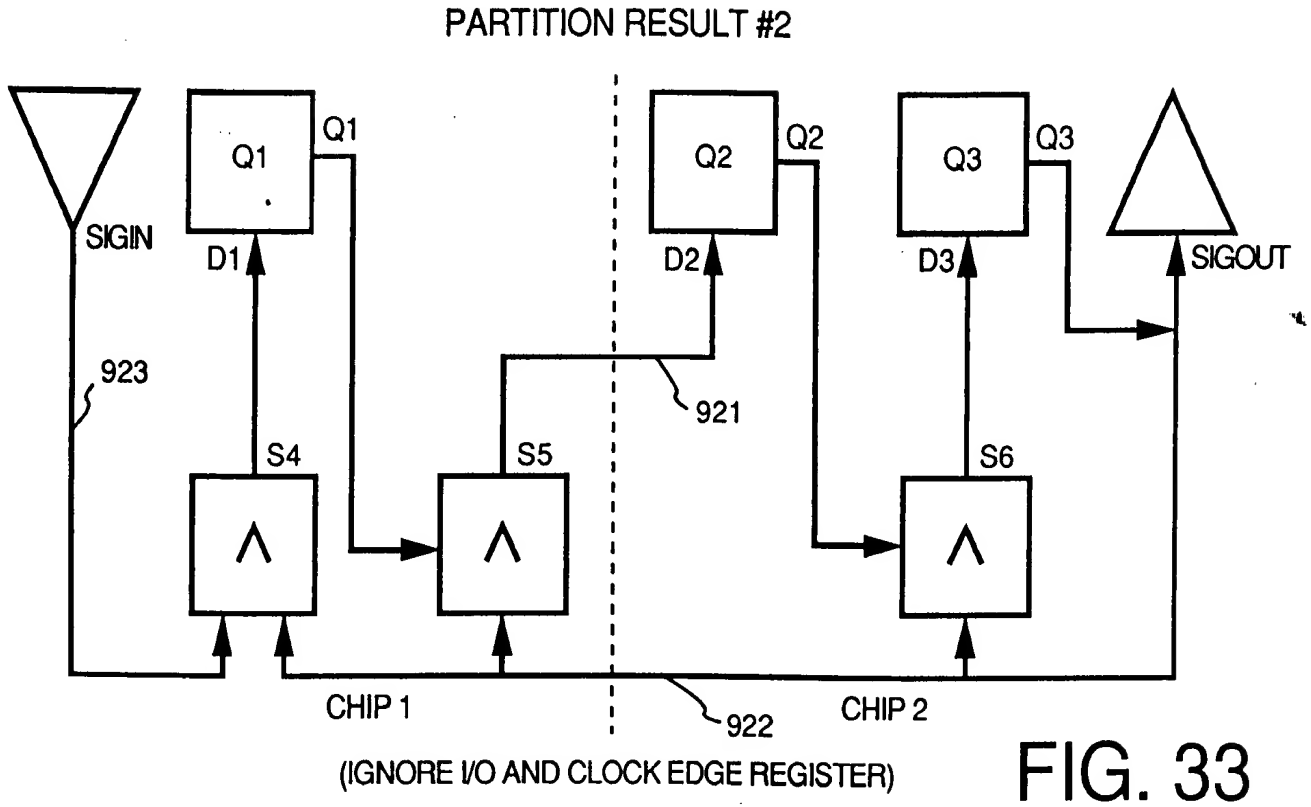


FIG. 33

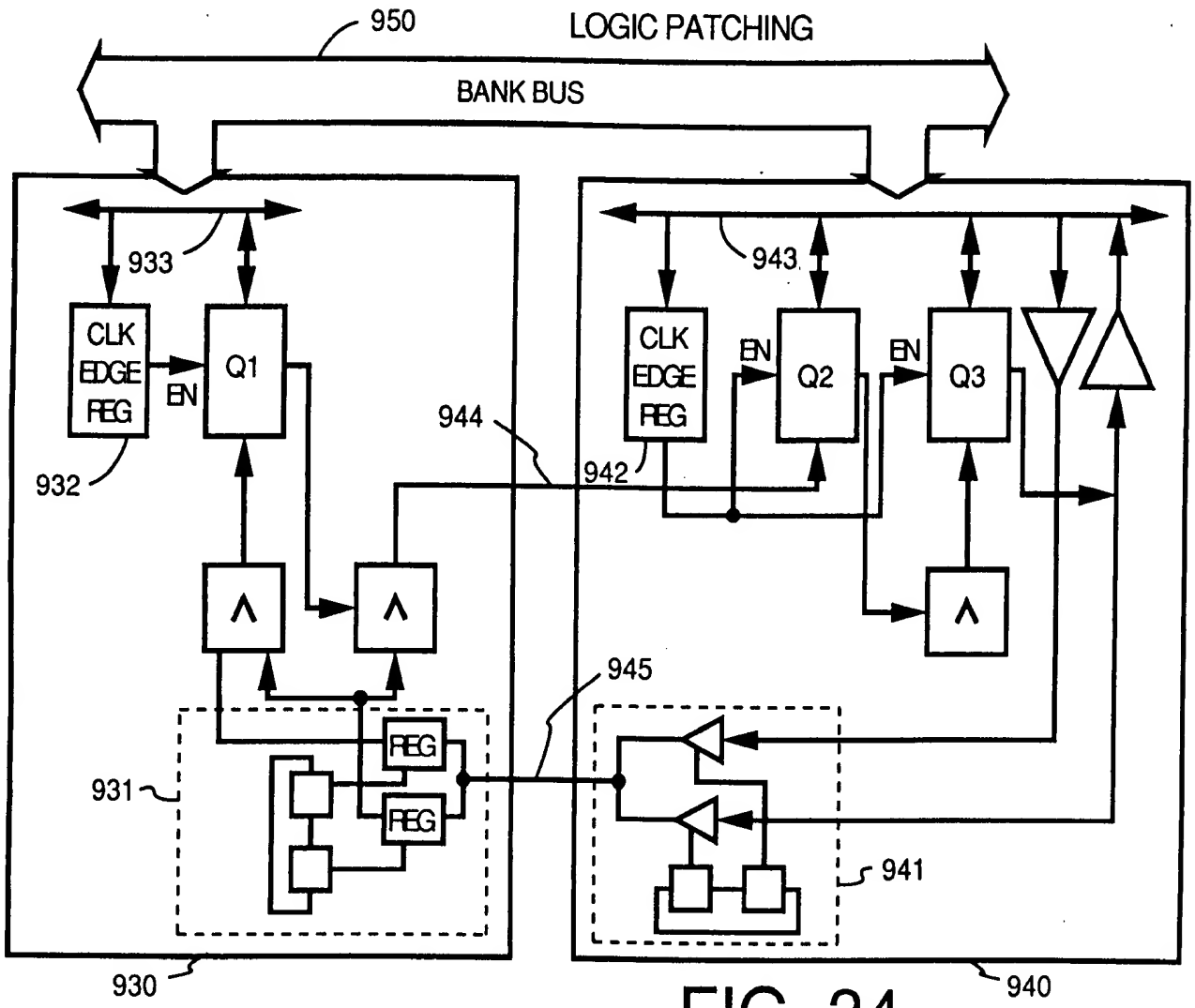


FIG. 34

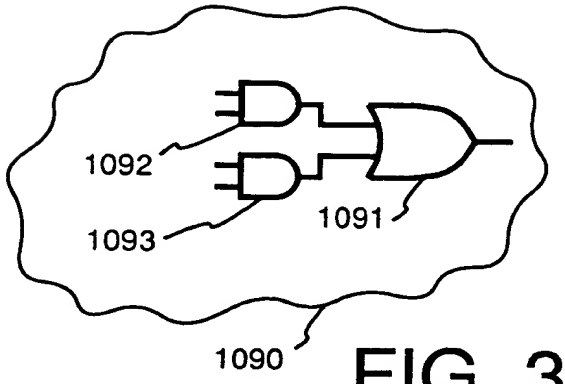


FIG. 35a

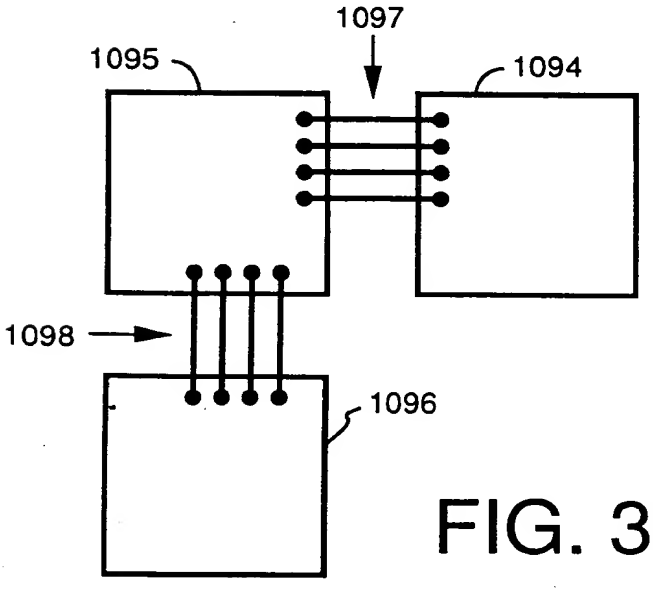


FIG. 35b

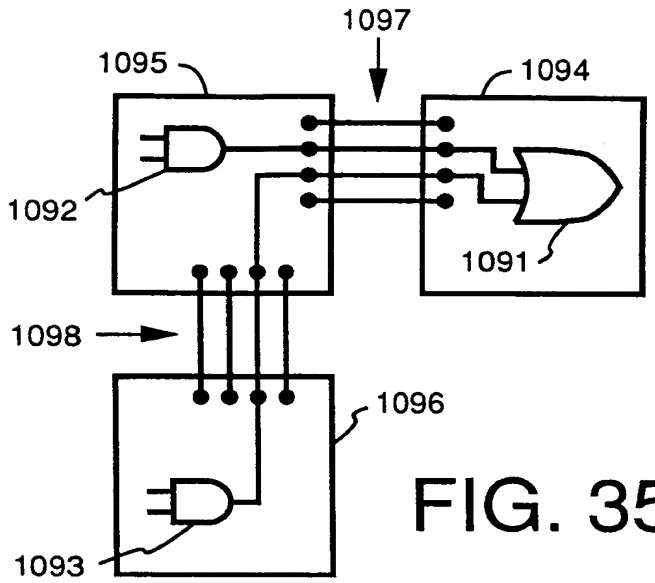


FIG. 35c

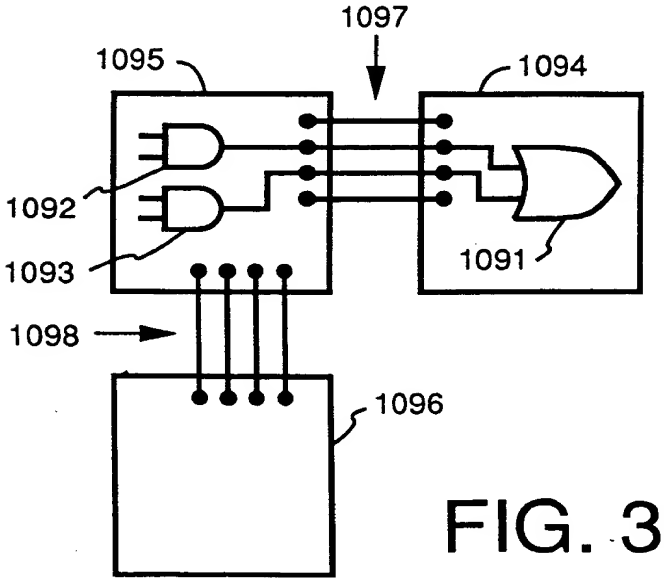
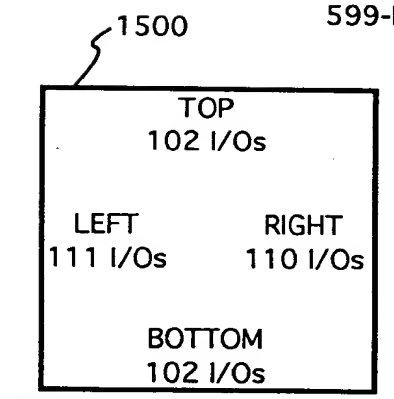


FIG. 35d

I/O PIN OVERVIEW OF
FPGA LOGIC DEVICE

FPGA : 10K130V, 10K250V WITH
599-PIN PGA PACKAGE



45 DEDICATED I/O PINS :

GCLK, FD_BUS[31..0], F_RD, F_WR,
DATAXSFR, SHIFTIN, SHIFTOUT,
SPACE[2..0], EVAL, EV_REQ_N,
DEV_OE, DEV_CLRN

FIG. 36

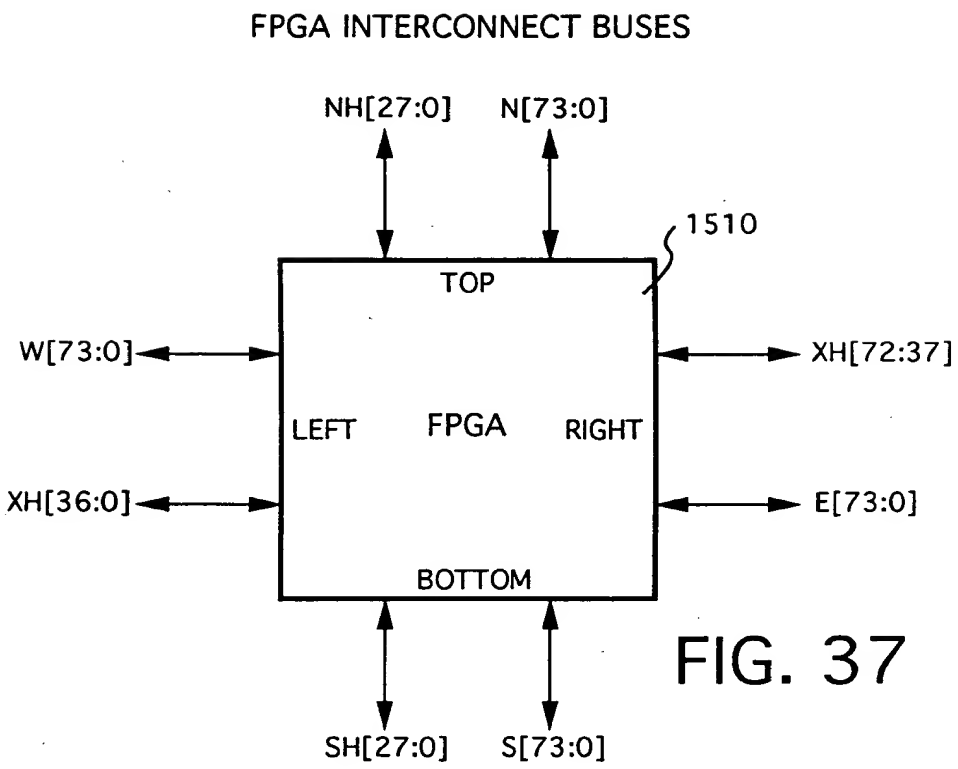


FIG. 37

BOARD CONNECTION - SIDE VIEW

DUAL - BOARD CONFIGURATION

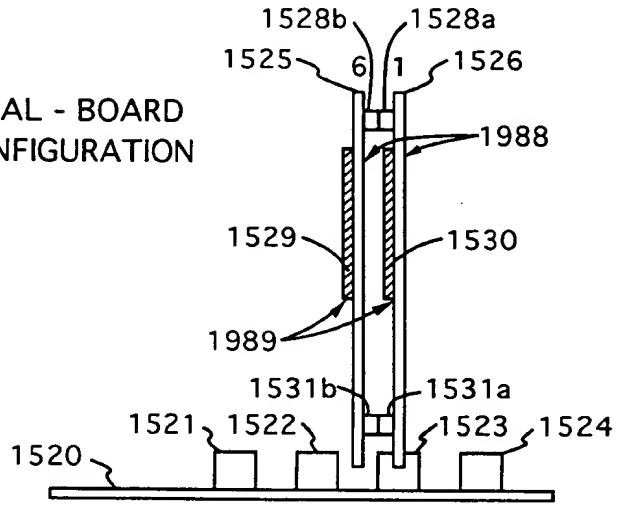


FIG. 38(a)

SIX BOARD CONFIGURATION

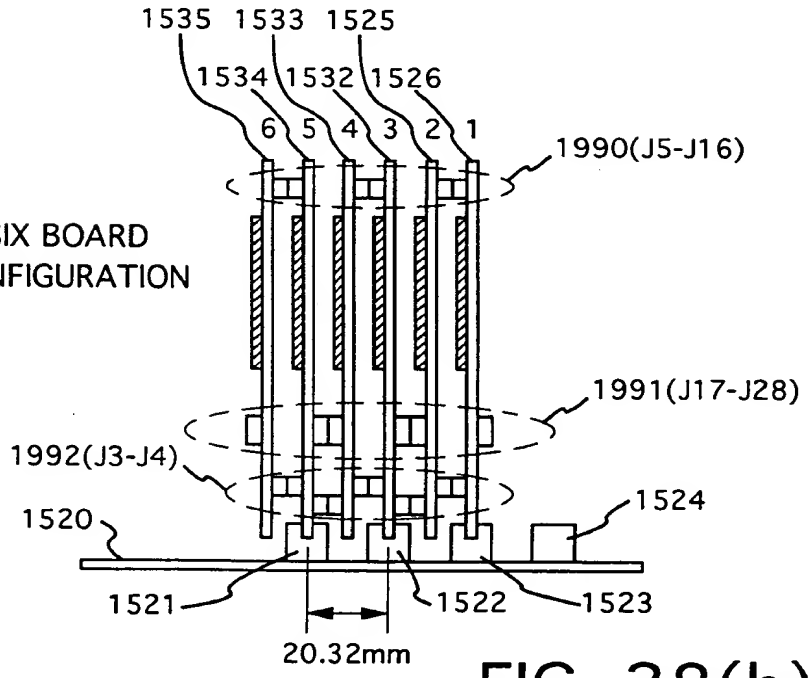


FIG. 38(b)

SIX-BOARD CONFIGURATION DIRECT-NEIGHBOR AND ONE-HOP FPGA ARRAY-X TORUS, Y MESH

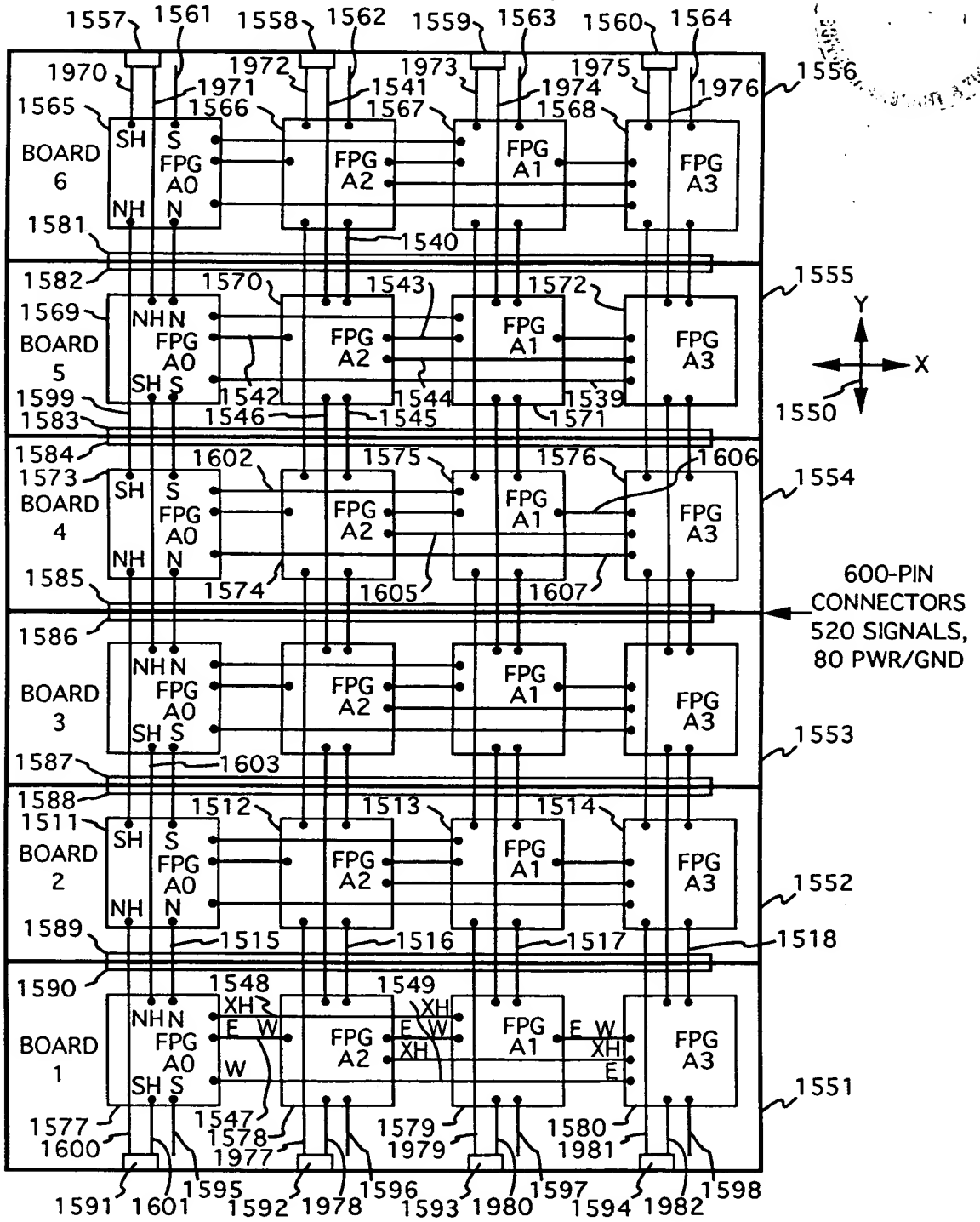


FIG. 39

FPGA ARRAY CONNECTION BETWEEN BOARDS

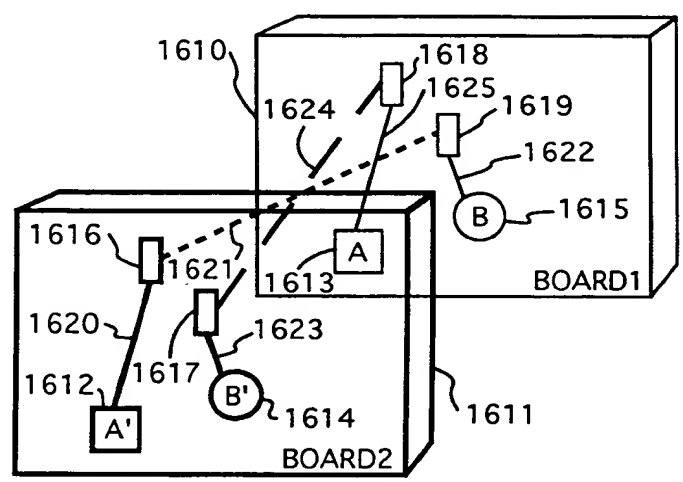


FIG. 40(a)

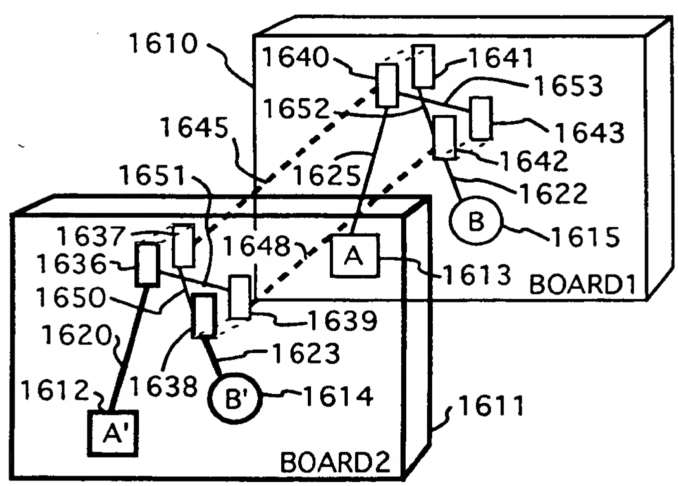


FIG. 40(b)

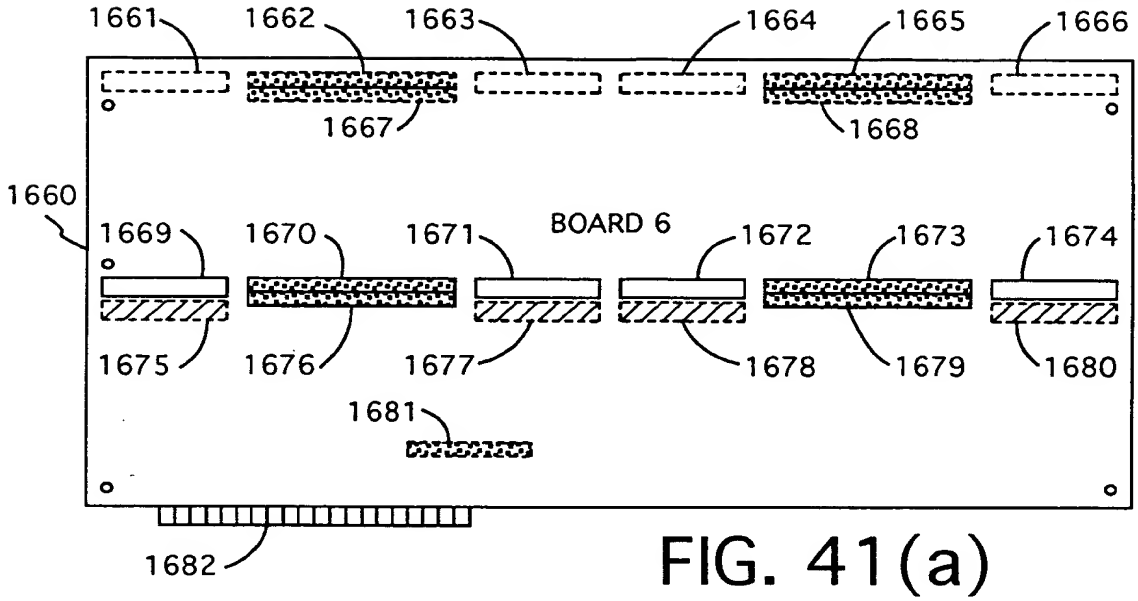


FIG. 41(a)

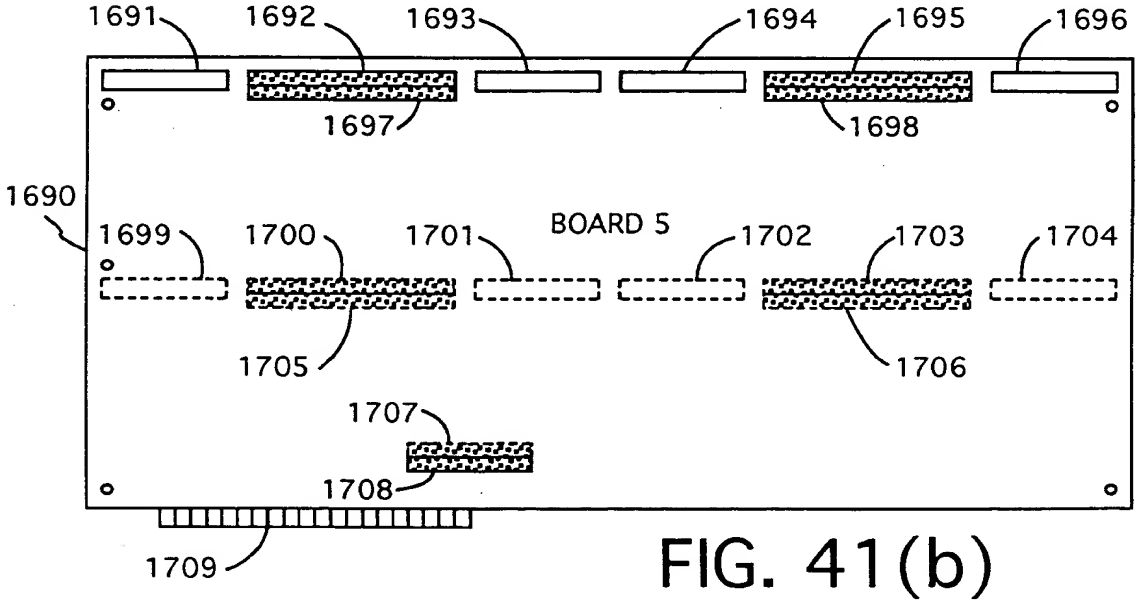


FIG. 41(b)

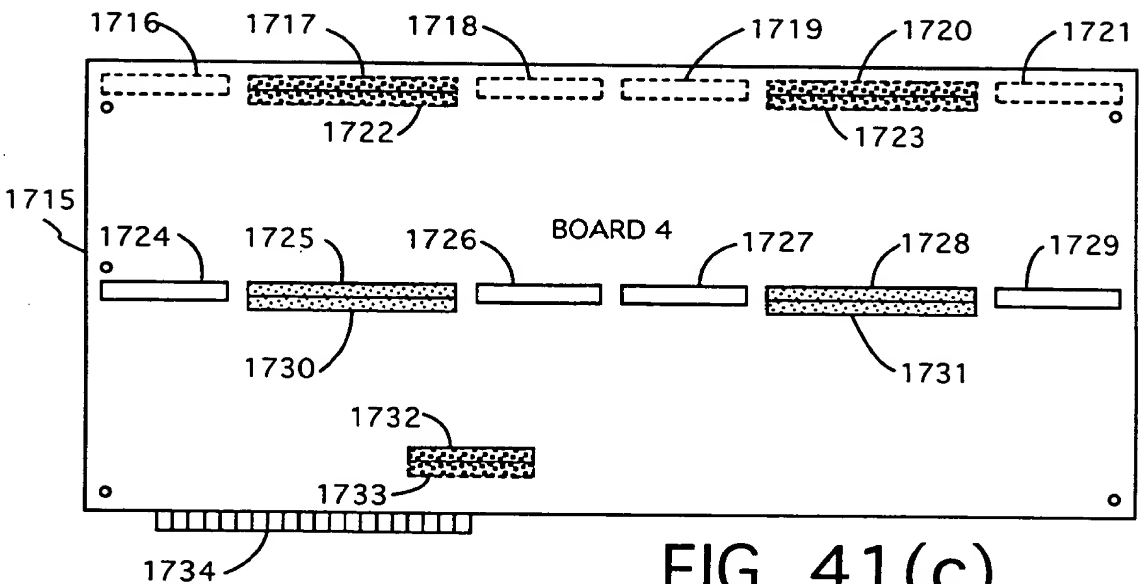


FIG. 41(c)

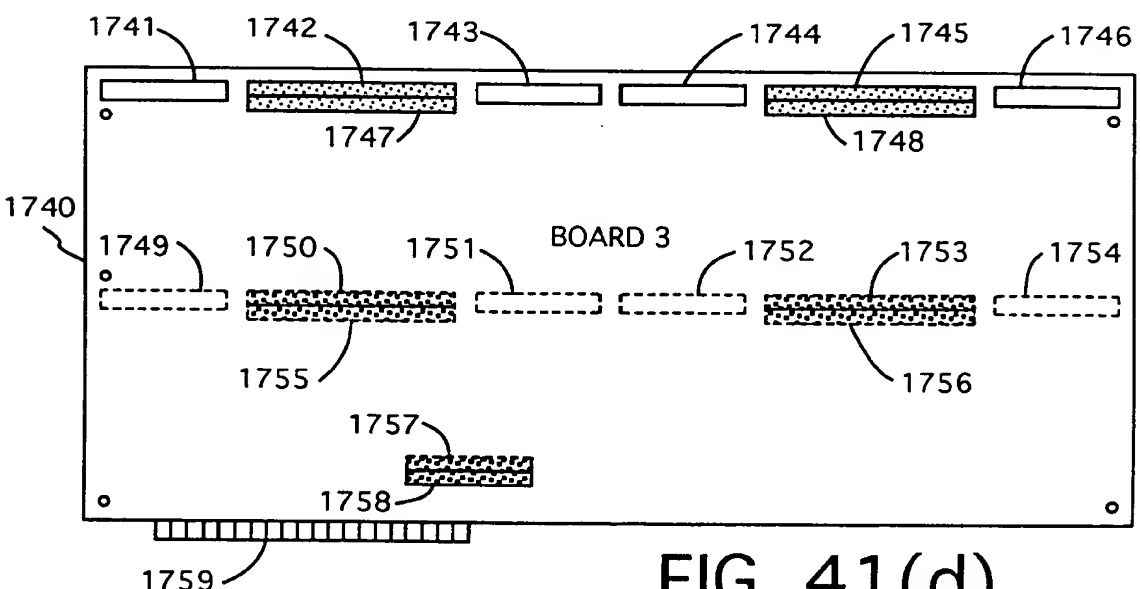


FIG. 41(d)

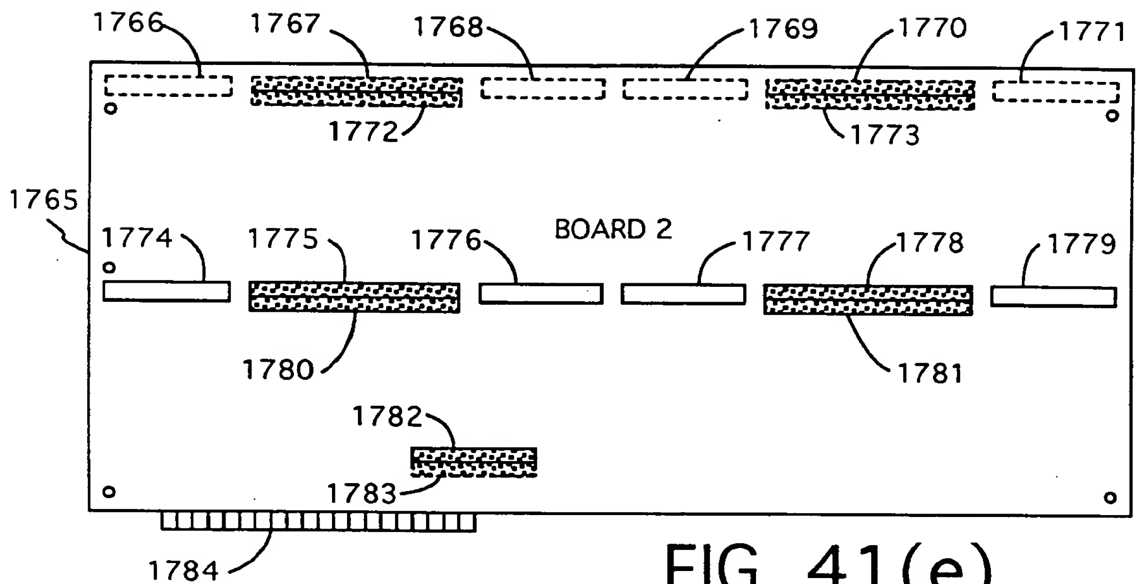


FIG. 41(e)

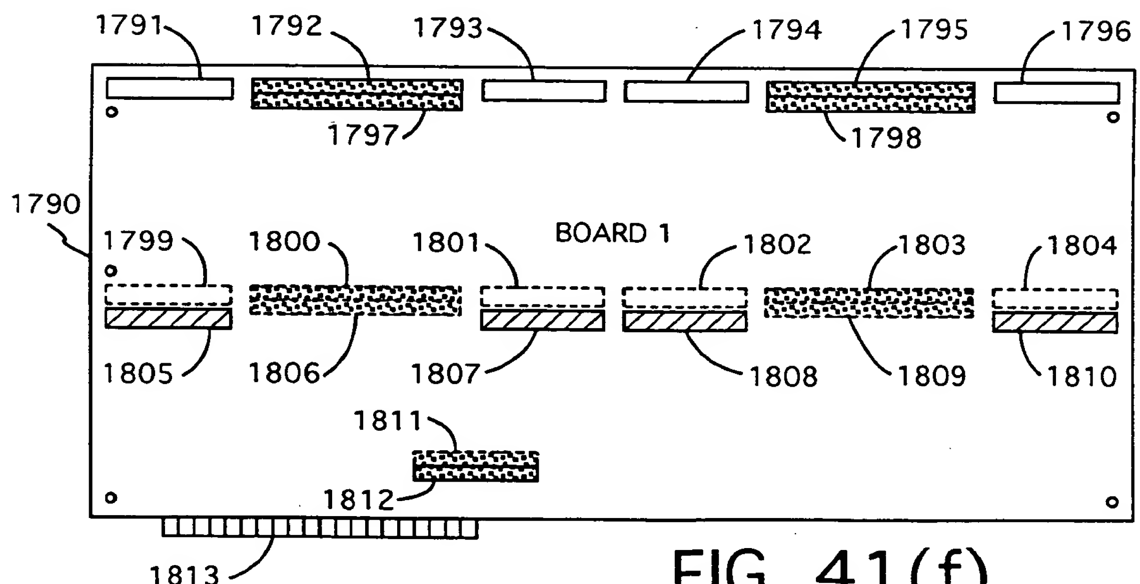
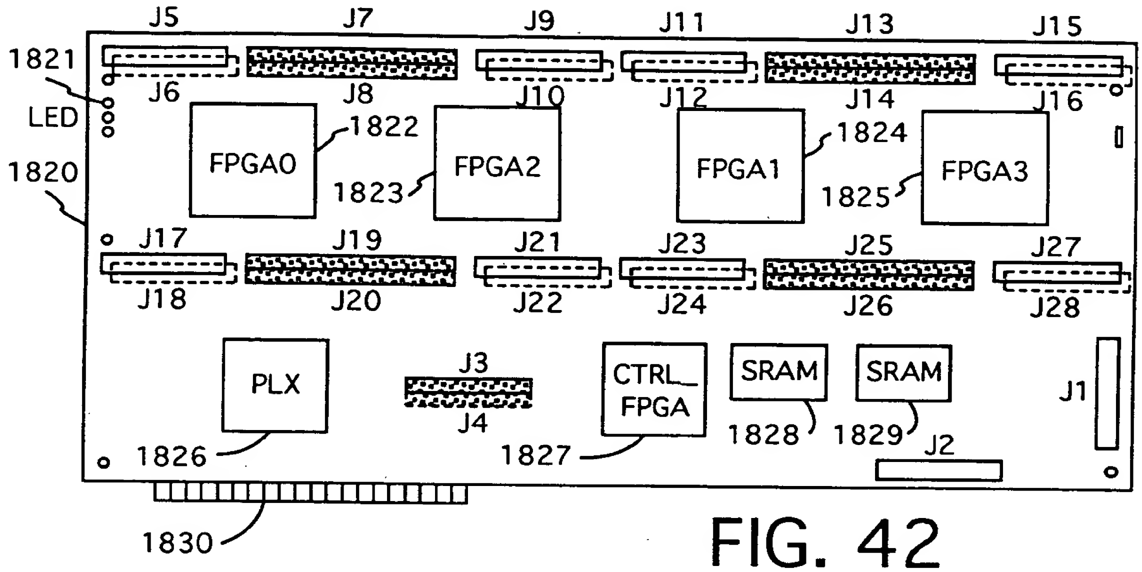
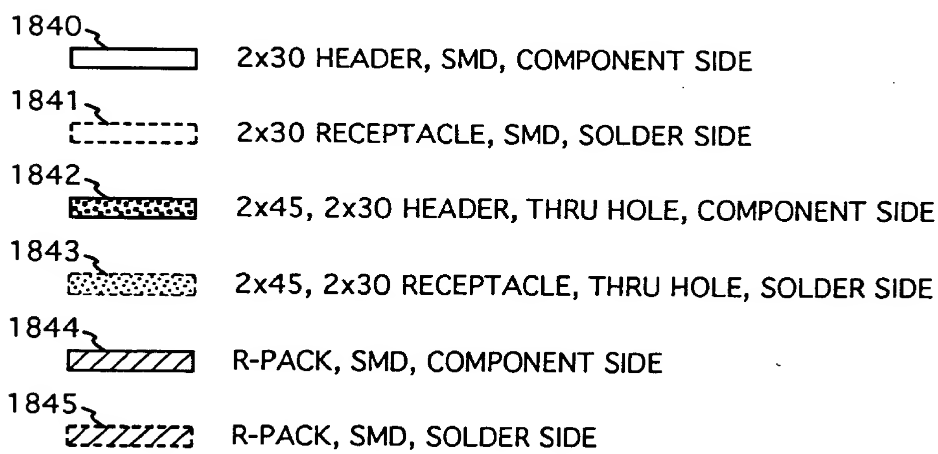


FIG. 41(f)



J5
J7
J9
J11
J13
J15
1821
J6
J8
J10
J12
J14
J16
LED
FPGA0
1822
FPGA2
FPGA1
1824
FPGA3
1820
1823
1825
J17
J19
J21
J23
J25
J27
J18
J20
J22
J24
J26
J28
PLX
J3
CTRL_ FPGA
SRAM
SRAM
J1
J4
1828
1829
J2
1826
1827
1830

FIG. 42



1840
2x30 HEADER, SMD, COMPONENT SIDE
1841
2x30 RECEPTACLE, SMD, SOLDER SIDE
1842
2x45, 2x30 HEADER, THRU HOLE, COMPONENT SIDE
1843
2x45, 2x30 RECEPTACLE, THRU HOLE, SOLDER SIDE
1844
R-PACK, SMD, COMPONENT SIDE
1845
R-PACK, SMD, SOLDER SIDE

FIG. 43

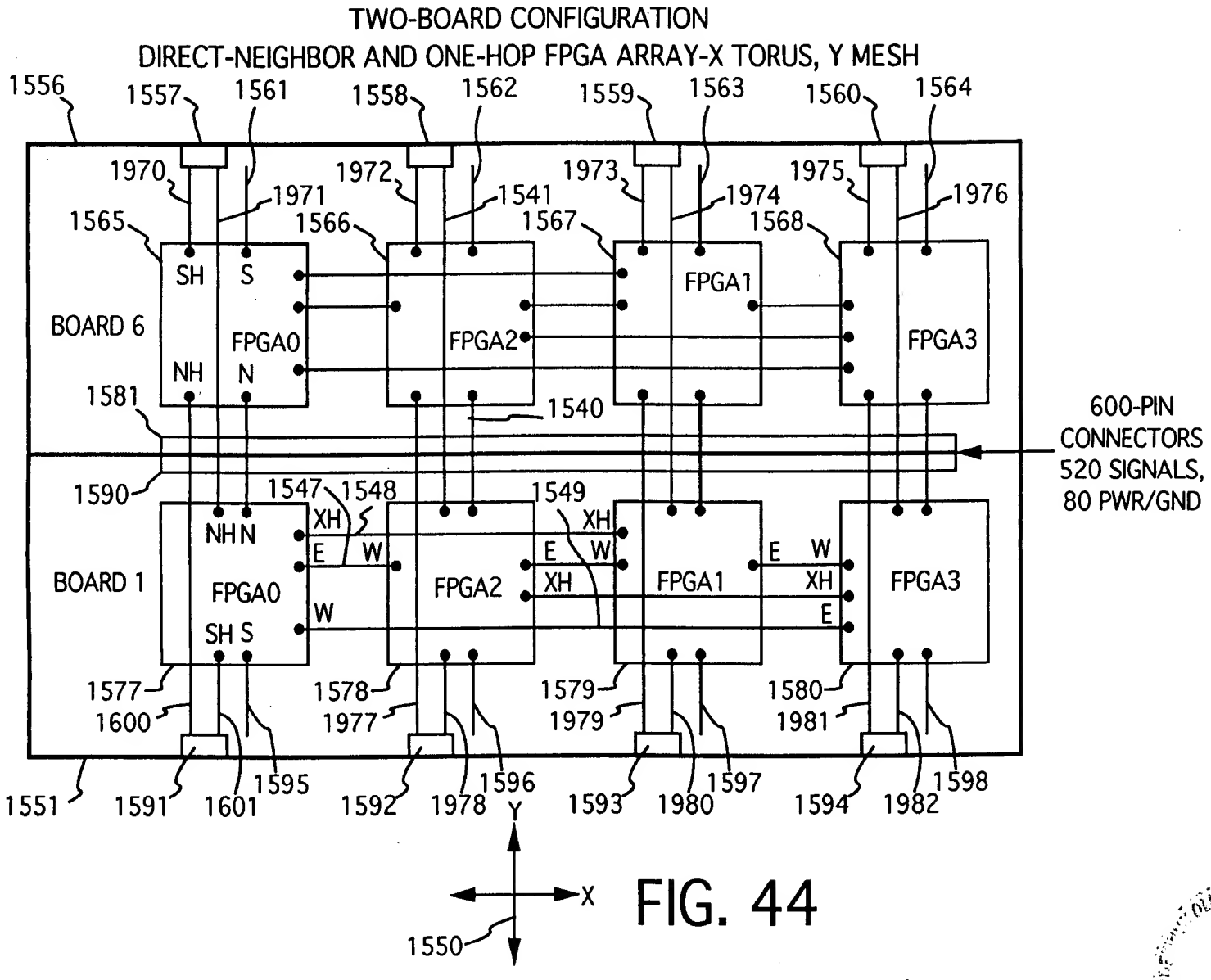

TWO-BOARD CONFIGURATION
DIRECT-NEIGHBOR AND ONE-HOP FPGA ARRAY-X TORUS, Y MESH
1556
1557
1561
1558
1562
1559
1563
1560
1564
1970
1971
1972
1541
1973
1974
1975
1976
1565
1566
1567
1568
SH
S
FPGA1
BOARD 6
FPGA0
FPGA2
FPGA3
NH
N
1581
1540
600-PIN
CONNECTORS
520 SIGNALS,
80 PWR/GND
1590
1547
1548
1549
XH
NH N
E
W
BOARD 1
FPGA0
FPGA2
FPGA1
FPGA3
SH S
1577
1578
1579
1580
1600
1977
1979
1981
1551
1591
1601
1595
1592
1978
1596
1593
1980
1597
1594
1982
1598
Y
X
1550

FIG. 44

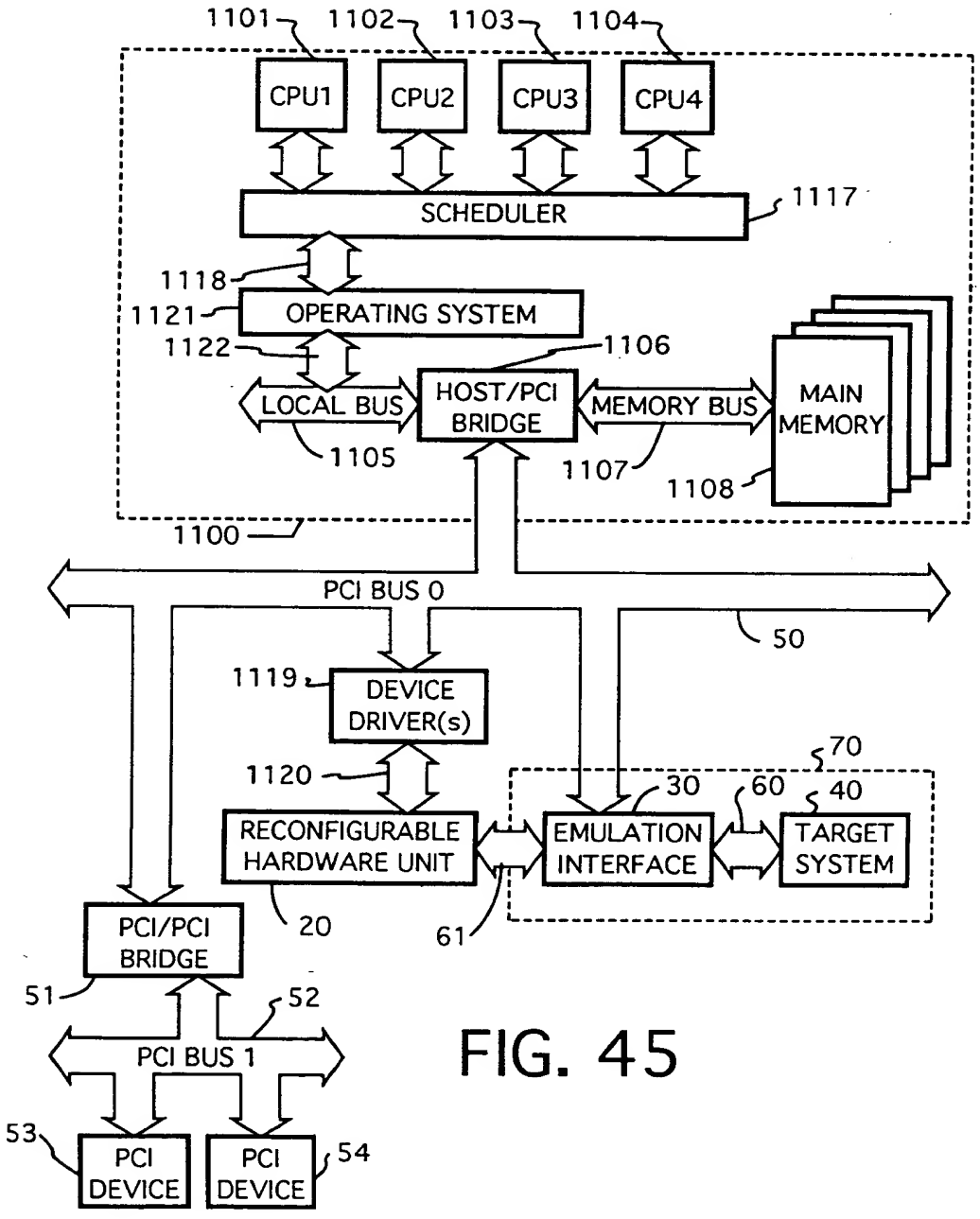


FIG. 45

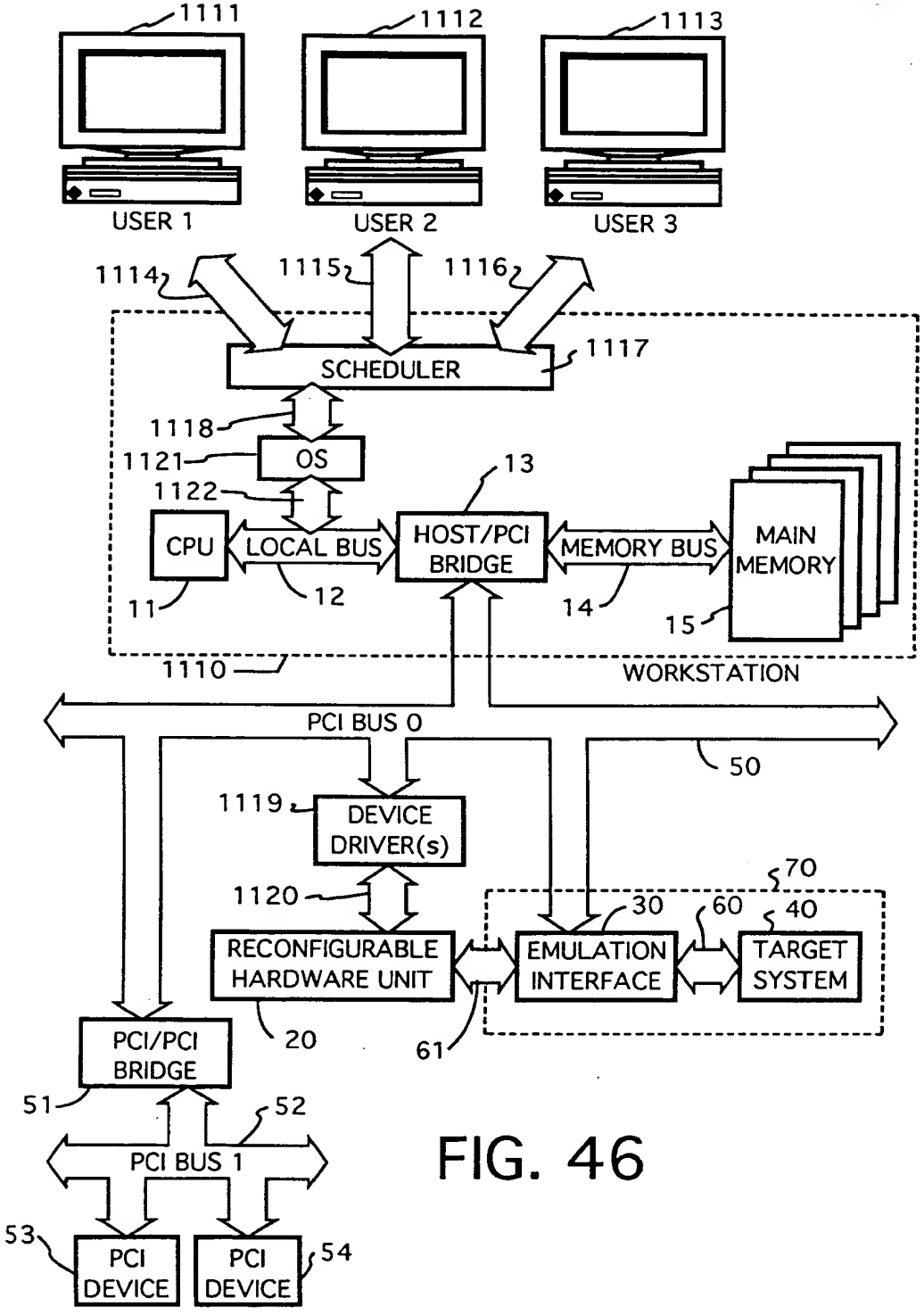

1111
1112
1113
USER 1
USER 2
USER 3
1114
1115
1116
SCHEDULER
1117
1118
OS
1121
1122
13
CPU
LOCAL BUS
HOST/PCI BRIDGE
MEMORY BUS
MAIN MEMORY
11
12
14
15
1110
WORKSTATION
PCI BUS 0
50
1119
DEVICE DRIVER(s)
1120
70
30
60
40
RECONFIGURABLE HARDWARE UNIT
EMULATION INTERFACE
TARGET SYSTEM
20
61
PCI/PCI BRIDGE
51
52
PCI BUS 1
53
PCI DEVICE
PCI DEVICE
54

FIG. 46

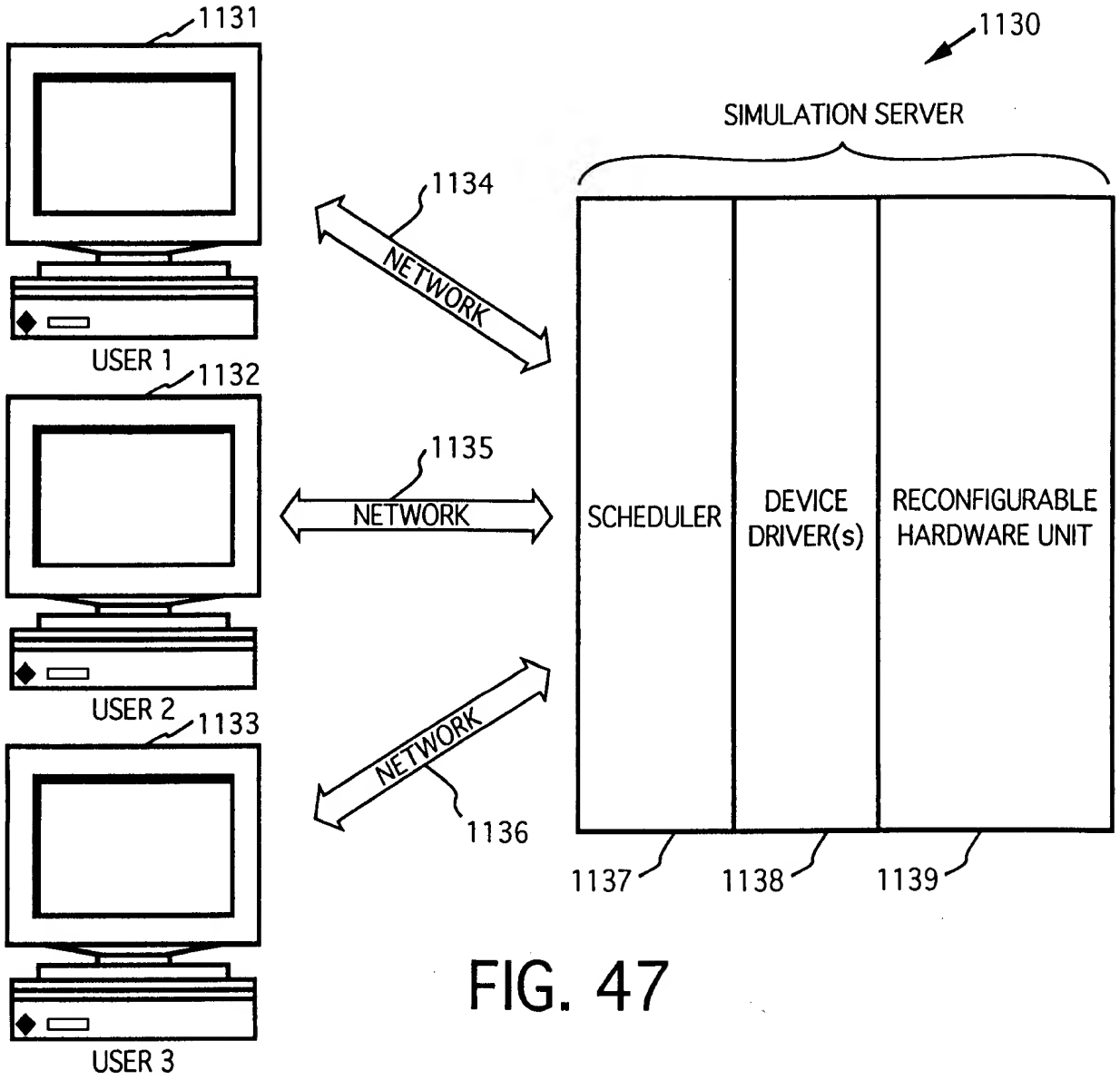

1131
1130
SIMULATION SERVER
1134
NETWORK
USER 1
1132
1135
NETWORK
SCHEDULER
DEVICE DRIVER(s)
RECONFIGURABLE HARDWARE UNIT
USER 2
1133
NETWORK
1136
1137
1138
1139
USER 3

FIG. 47

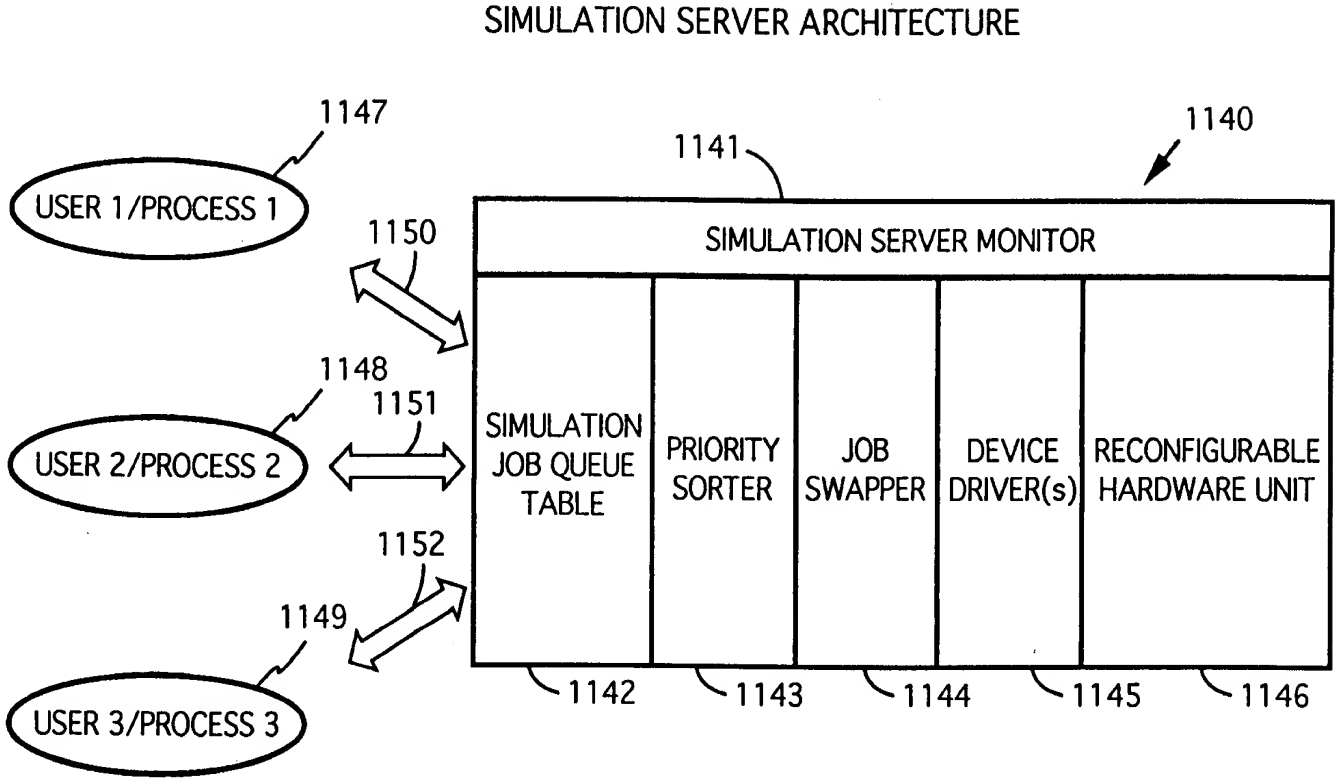

SIMULATION SERVER ARCHITECTURE
1147
1140
1141
USER 1/PROCESS 1
1150
SIMULATION SERVER MONITOR
1148
1151
USER 2/PROCESS 2
SIMULATION JOB QUEUE TABLE
PRIORITY SORTER
JOB SWAPPER
DEVICE DRIVER(s)
RECONFIGURABLE HARDWARE UNIT
1152
1149
USER 3/PROCESS 3
1142
1143
1144
1145
1146

FIG. 48

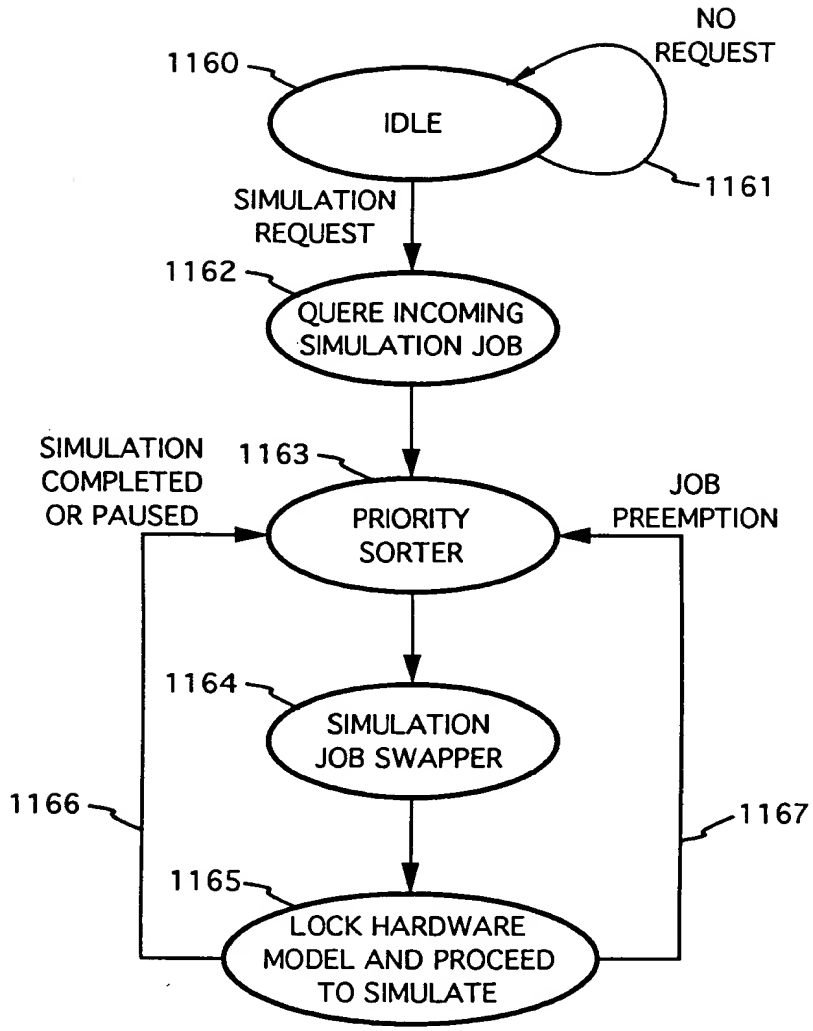


FIG. 49

JOB SWAPPER

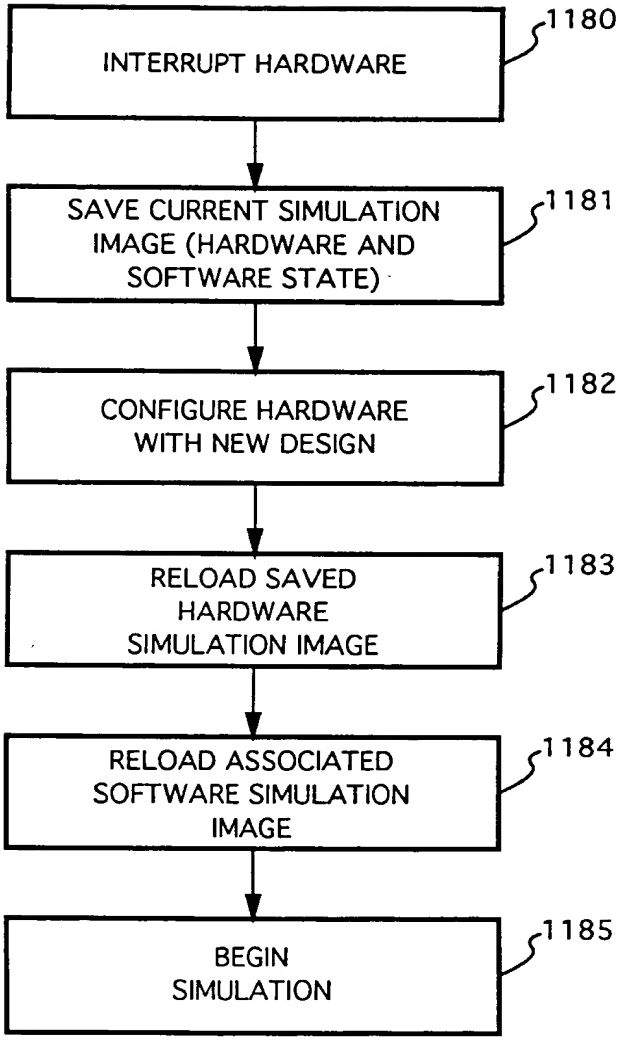


FIG. 50

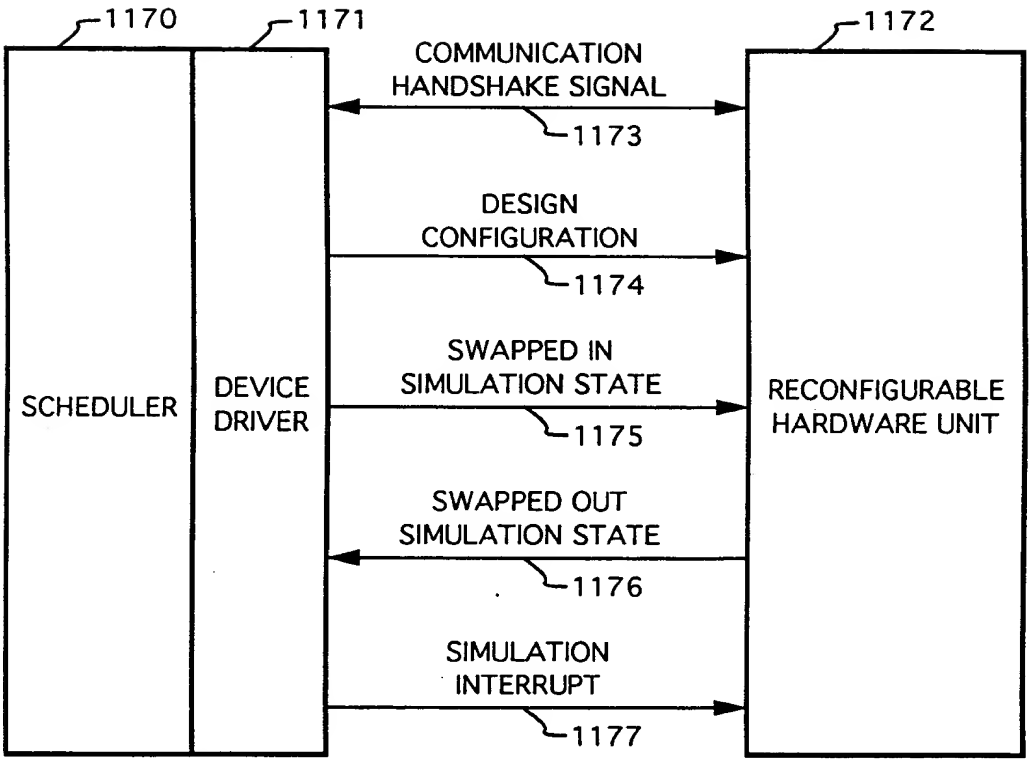


FIG. 51

PRIORITY I { JOB A, JOB B

PRIORITY II { JOB C, JOB D

TIME-SHARED HARDWARE USAGE:

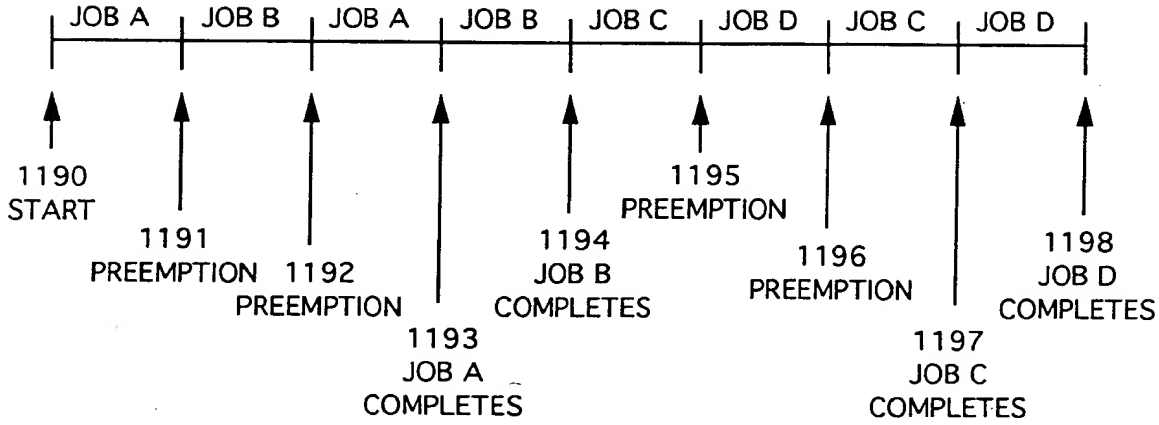


FIG. 52

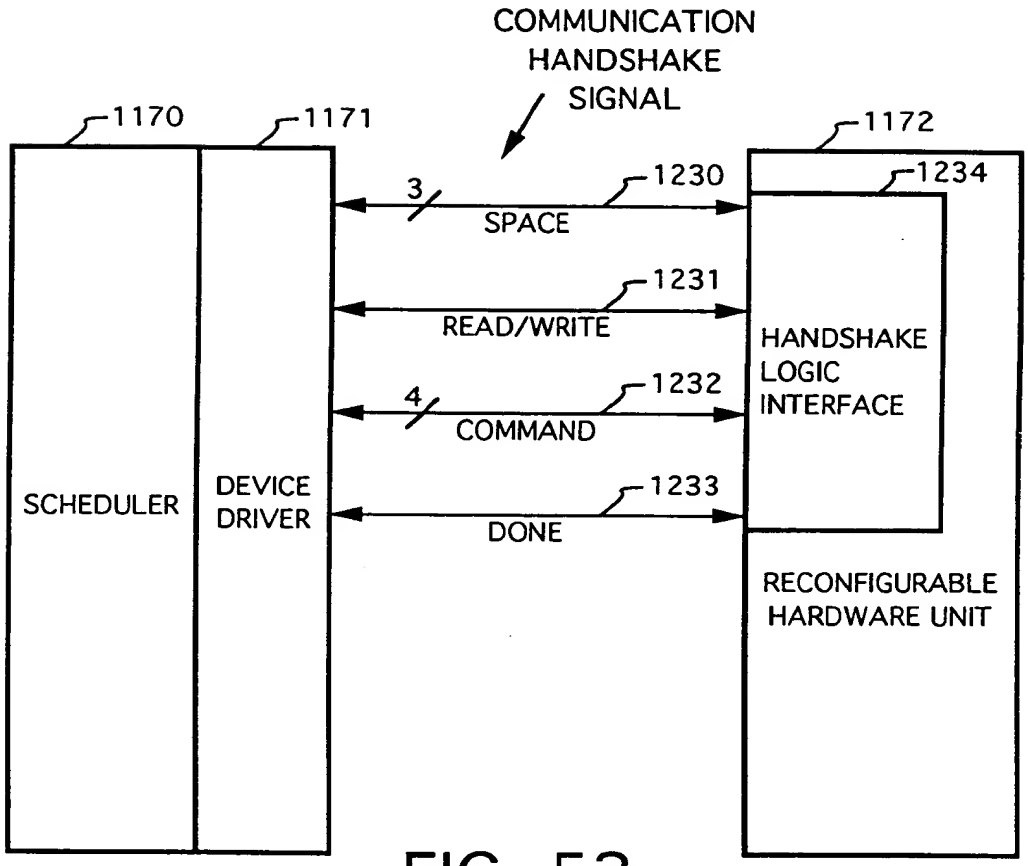


FIG. 53

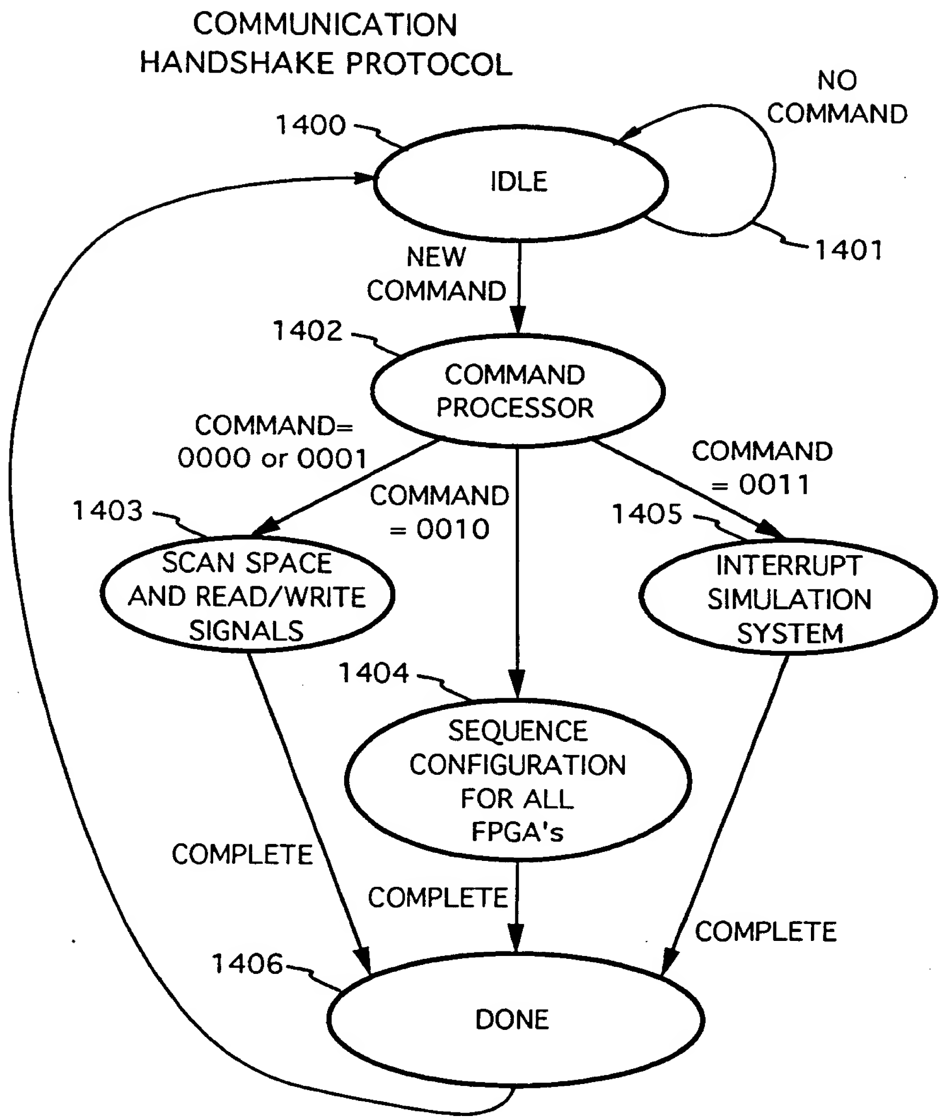

COMMUNICATION
HANDSHAKE PROTOCOL
NO
COMMAND
1400
IDLE
1401
NEW
COMMAND
1402
COMMAND
PROCESSOR
COMMAND=
0000 or 0001
COMMAND
= 0010
COMMAND
= 0011
1403
SCAN SPACE
AND READ/WRITE
SIGNALS
1405
INTERRUPT
SIMULATION
SYSTEM
1404
SEQUENCE
CONFIGURATION
FOR ALL
FPGA's
COMPLETE
COMPLETE
COMPLETE
1406
DONE

FIG. 54

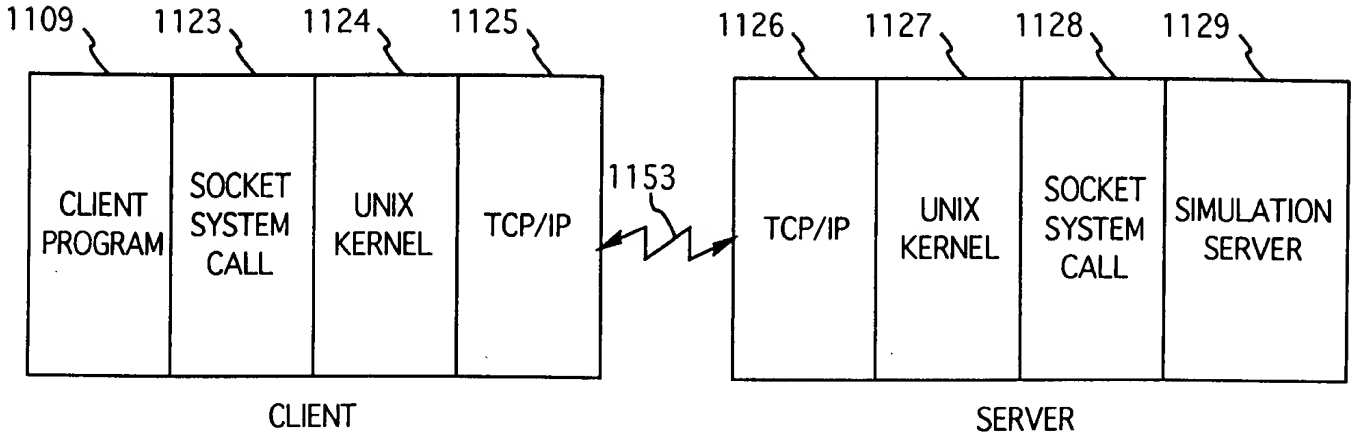
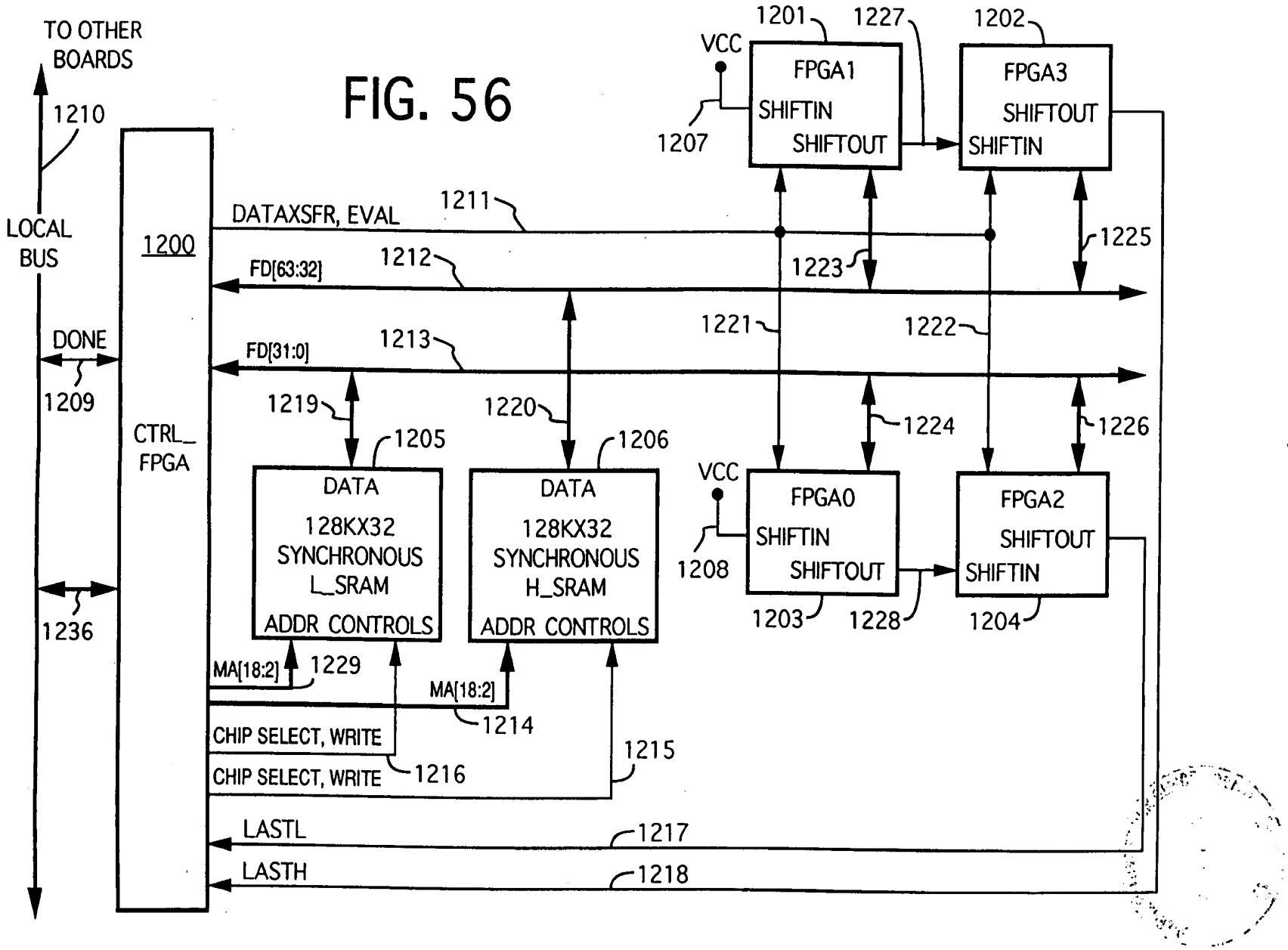

1109
1123
1124
1125
1126
1127
1128
1129
CLIENT PROGRAM
SOCKET SYSTEM CALL
UNIX KERNEL
TCP/IP
1153
TCP/IP
UNIX KERNEL
SOCKET SYSTEM CALL
SIMULATION SERVER
CLIENT
SERVER

FIG. 55

FIG. 56


TO OTHER BOARDS
1210
LOCAL BUS
1200
CTRL_ FPGA
DONE
1209
1236
DATAXSFR, EVAL
1211
FD[63:32]
1212
FD[31:0]
1213
1219
1220
1205
1206
DATA 128KX32 SYNCHRONOUS L_SRAM ADDR CONTROLS
DATA 128KX32 SYNCHRONOUS H_SRAM ADDR CONTROLS
MA[18:2]
1229
MA[18:2]
1214
CHIP SELECT, WRITE
1216
CHIP SELECT, WRITE
1215
LASTL
1217
LASTH
1218
VCC
1207
1201
1227
FPGA1 SHIFTIN SHIFTOUT
1202
FPGA3 SHIFTOUT SHIFTIN
1223
1225
1221
1222
1224
1226
VCC
1208
FPGA0 SHIFTIN SHIFTOUT
FPGA2 SHIFTOUT SHIFTIN
1203
1228
1204

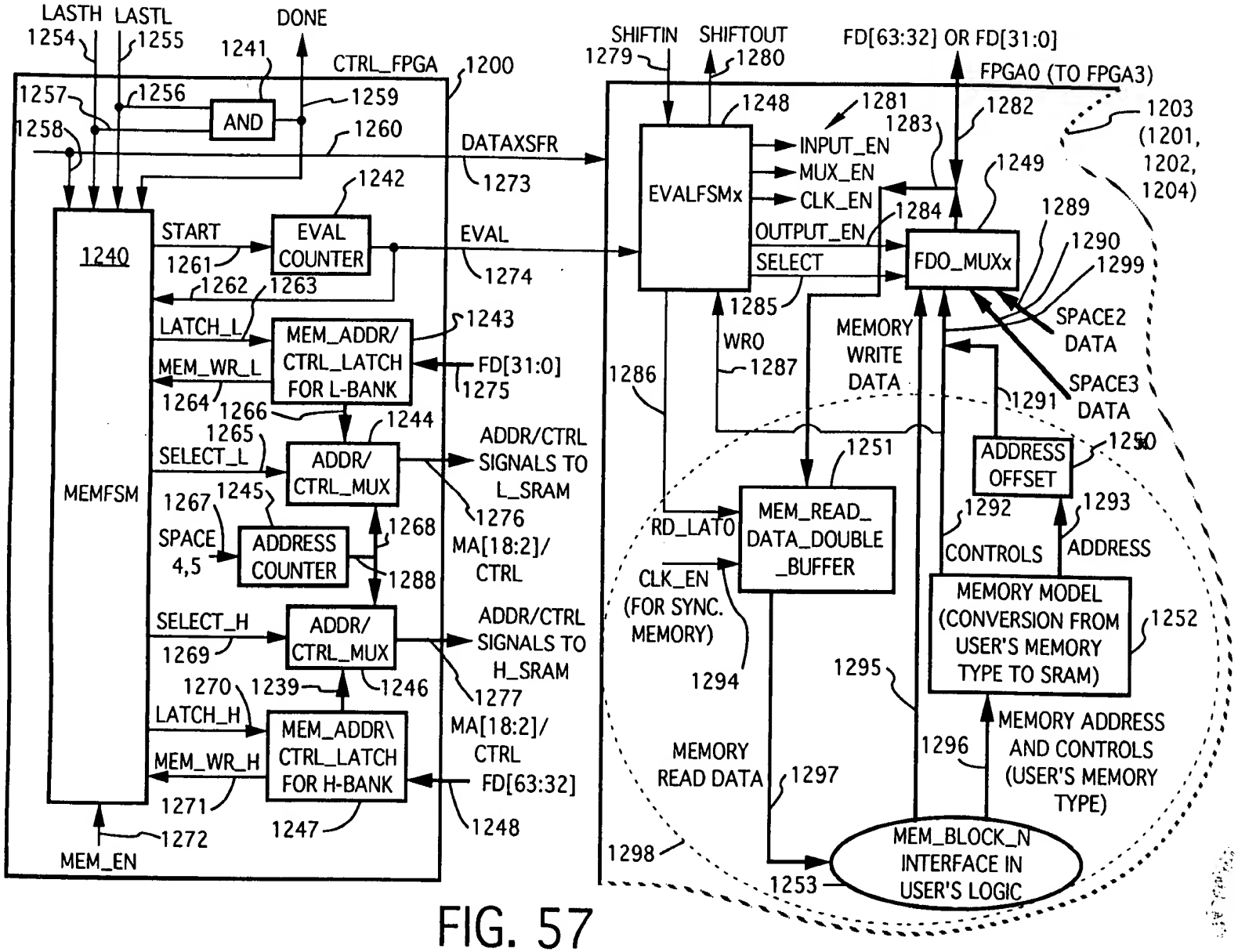

LASTH
1254
LASTL
1255
DONE
1241
CTRL_FPGA
1200
1257
1256
AND
1259
1258
1260
DATAXSFR
1273
1242
1240
START
1261
EVAL
COUNTER
EVAL
1274
1262
1263
LATCH_L
MEM_ADDR/
CTRL_LATCH
FOR L-BANK
1243
MEM_WR_L
FD[31:0]
1275
1264
1266
1244
1265
SELECT_L
ADDR/
CTRL_MUX
ADDR/CTRL
SIGNALS TO
L_SRAM
1276
MEMFSM
1267
1245
SPACE
4,5
ADDRESS
COUNTER
1268
MA[18:2]/
CTRL
1288
SELECT_H
1269
ADDR/
CTRL_MUX
ADDR/CTRL
SIGNALS TO
H_SRAM
1277
1270
1239
1246
LATCH_H
MEM_ADDR\
CTRL_LATCH
FOR H-BANK
MA[18:2]/
CTRL
MEM_WR_H
FD[63:32]
1271
1247
1248
1272
MEM_EN
SHIFTIN
1279
SHIFTOUT
1280
FD[63:32] OR FD[31:0]
FPGA0 (TO FPGA3)
1248
1281
1282
1203
(1201,
1202,
1204)
1283
EVALFSMx
INPUT_EN
MUX_EN
CLK_EN
OUTPUT_EN
SELECT
1284
1249
1289
1290
1299
FDO_MUXx
1285
WRO
1287
MEMORY
WRITE
DATA
SPACE2
DATA
SPACE3
DATA
1286
1291
1250
ADDRESS
OFFSET
1251
MEM_READ_
DATA_DOUBLE
_BUFFER
RD_LAT0
1292
1293
CONTROLS
ADDRESS
CLK_EN
(FOR SYNC.
MEMORY)
MEMORY MODEL
(CONVERSION FROM
USER'S MEMORY
TYPE TO SRAM)
1252
1294
1295
MEMORY ADDRESS
AND CONTROLS
(USER'S MEMORY
TYPE)
1296
MEMORY
READ DATA
1297
1298
MEM_BLOCK_N
INTERFACE IN
USER'S LOGIC
1253

FIG. 57

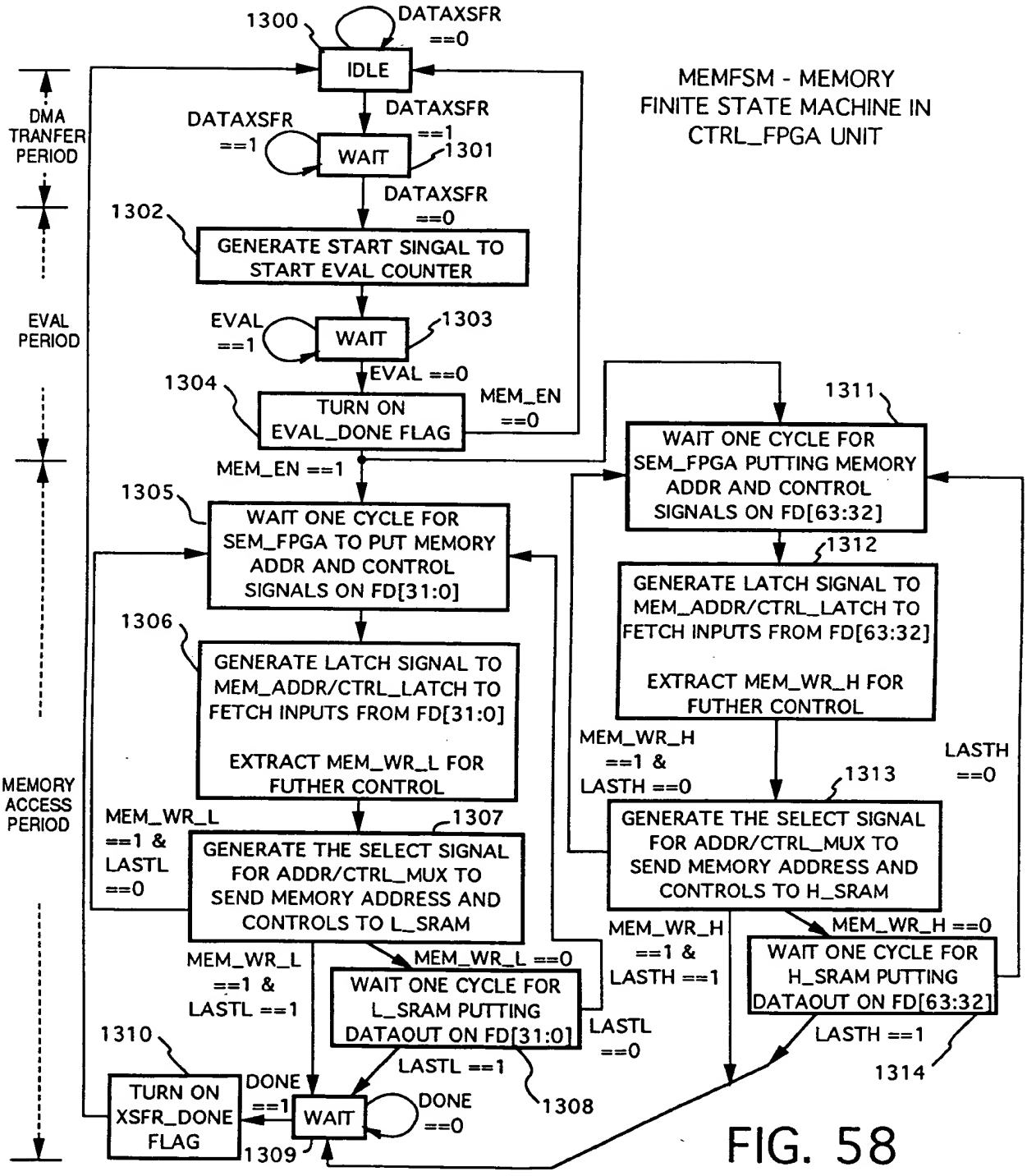

MEMFSM - MEMORY FINITE STATE MACHINE IN CTRL_FPGA UNIT
DMA TRANFER PERIOD
EVAL PERIOD
MEMORY ACCESS PERIOD
1300
DATAXSFR ==0
IDLE
DATAXSFR ==1
DATAXSFR ==1
WAIT
1301
DATAXSFR ==0
1302
GENERATE START SINGAL TO START EVAL COUNTER
EVAL ==1
WAIT
1303
EVAL ==0
1304
TURN ON EVAL_DONE FLAG
MEM_EN ==0
MEM_EN ==1
1305
WAIT ONE CYCLE FOR SEM_FPGA TO PUT MEMORY ADDR AND CONTROL SIGNALS ON FD[31:0]
1306
GENERATE LATCH SIGNAL TO MEM_ADDR/CTRL_LATCH TO FETCH INPUTS FROM FD[31:0]
EXTRACT MEM_WR_L FOR FUTHER CONTROL
MEM_WR_L ==1 & LASTL ==0
1307
GENERATE THE SELECT SIGNAL FOR ADDR/CTRL_MUX TO SEND MEMORY ADDRESS AND CONTROLS TO L_SRAM
MEM_WR_L ==1 & LASTL ==1
MEM_WR_L ==0
WAIT ONE CYCLE FOR L_SRAM PUTTING DATAOUT ON FD[31:0]
LASTL ==0
LASTL ==1
1308
1310
TURN ON XSFR_DONE FLAG
DONE ==1
WAIT
DONE ==0
1309
1311
WAIT ONE CYCLE FOR SEM_FPGA PUTTING MEMORY ADDR AND CONTROL SIGNALS ON FD[63:32]
1312
GENERATE LATCH SIGNAL TO MEM_ADDR/CTRL_LATCH TO FETCH INPUTS FROM FD[63:32]
EXTRACT MEM_WR_H FOR FUTHER CONTROL
MEM_WR_H ==1 & LASTH ==0
1313
GENERATE THE SELECT SIGNAL FOR ADDR/CTRL_MUX TO SEND MEMORY ADDRESS AND CONTROLS TO H_SRAM
LASTH ==0
MEM_WR_H ==1 & LASTH ==1
MEM_WR_H ==0
WAIT ONE CYCLE FOR H_SRAM PUTTING DATAOUT ON FD[63:32]
LASTH ==1
1314

FIG. 58

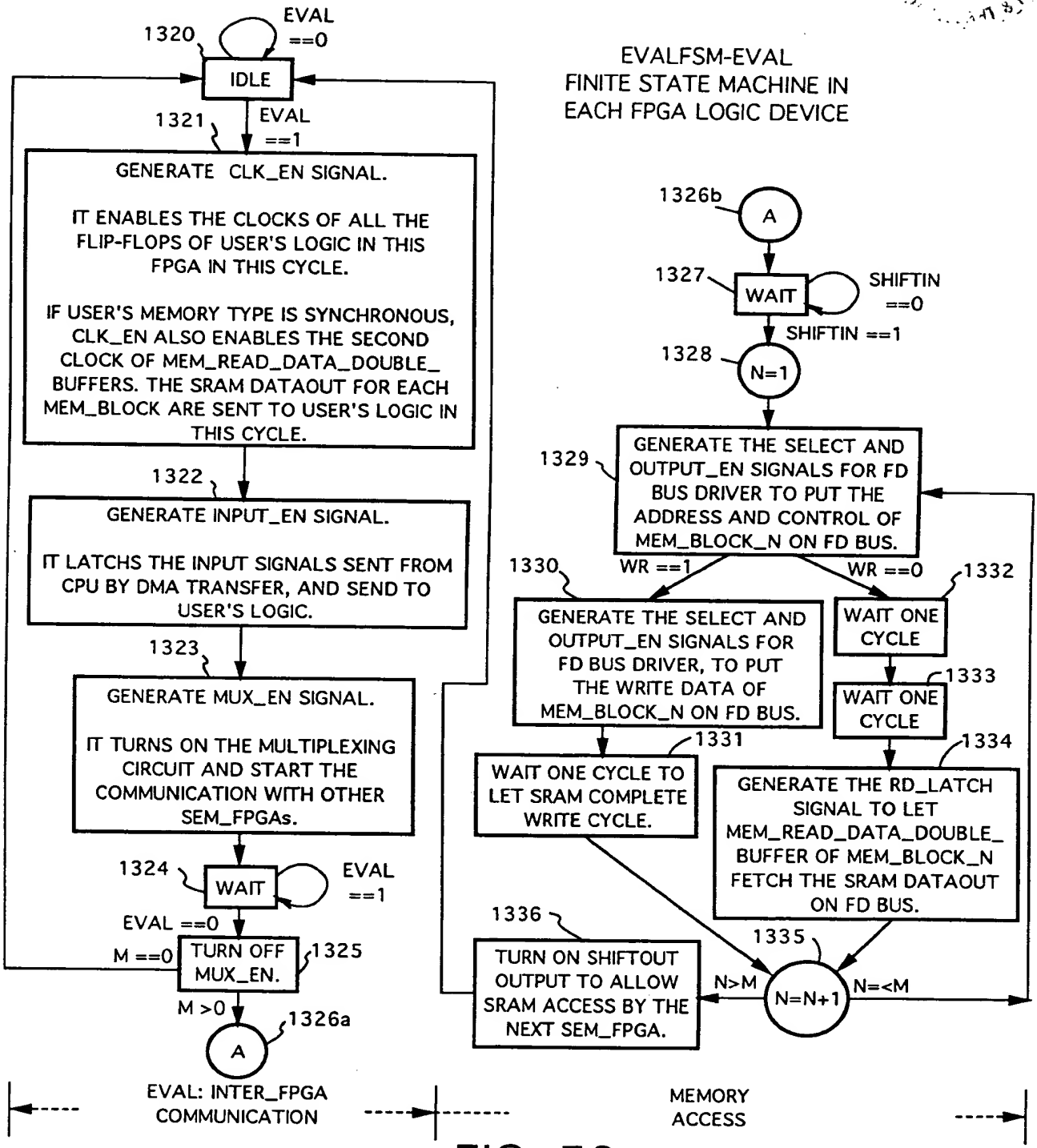

EVALFSM-EVAL
FINITE STATE MACHINE IN
EACH FPGA LOGIC DEVICE
1320
EVAL ==0
IDLE
1321
EVAL ==1
GENERATE CLK_EN SIGNAL.
IT ENABLES THE CLOCKS OF ALL THE FLIP-FLOPS OF USER'S LOGIC IN THIS FPGA IN THIS CYCLE.
IF USER'S MEMORY TYPE IS SYNCHRONOUS, CLK_EN ALSO ENABLES THE SECOND CLOCK OF MEM_READ_DATA_DOUBLE_ BUFFERS. THE SRAM DATAOUT FOR EACH MEM_BLOCK ARE SENT TO USER'S LOGIC IN THIS CYCLE.
1322
GENERATE INPUT_EN SIGNAL.
IT LATCHS THE INPUT SIGNALS SENT FROM CPU BY DMA TRANSFER, AND SEND TO USER'S LOGIC.
1323
GENERATE MUX_EN SIGNAL.
IT TURNS ON THE MULTIPLEXING CIRCUIT AND START THE COMMUNICATION WITH OTHER SEM_FPGAs.
1324
WAIT
EVAL ==1
EVAL ==0
M ==0
TURN OFF MUX_EN.
1325
M >0
A
1326a
1326b
A
1327
WAIT
SHIFTIN ==0
SHIFTIN ==1
1328
N=1
1329
GENERATE THE SELECT AND OUTPUT_EN SIGNALS FOR FD BUS DRIVER TO PUT THE ADDRESS AND CONTROL OF MEM_BLOCK_N ON FD BUS.
1330
WR ==1
WR ==0
1332
GENERATE THE SELECT AND OUTPUT_EN SIGNALS FOR FD BUS DRIVER, TO PUT THE WRITE DATA OF MEM_BLOCK_N ON FD BUS.
WAIT ONE CYCLE
1333
WAIT ONE CYCLE
1331
1334
WAIT ONE CYCLE TO LET SRAM COMPLETE WRITE CYCLE.
GENERATE THE RD_LATCH SIGNAL TO LET MEM_READ_DATA_DOUBLE_ BUFFER OF MEM_BLOCK_N FETCH THE SRAM DATAOUT ON FD BUS.
1336
1335
TURN ON SHIFTOUT OUTPUT TO ALLOW SRAM ACCESS BY THE NEXT SEM_FPGA.
N>M
N=N+1
N=<M
EVAL: INTER_FPGA COMMUNICATION
MEMORY ACCESS

FIG. 59

MEMORY READ DATA
DOUBLE BUFFER

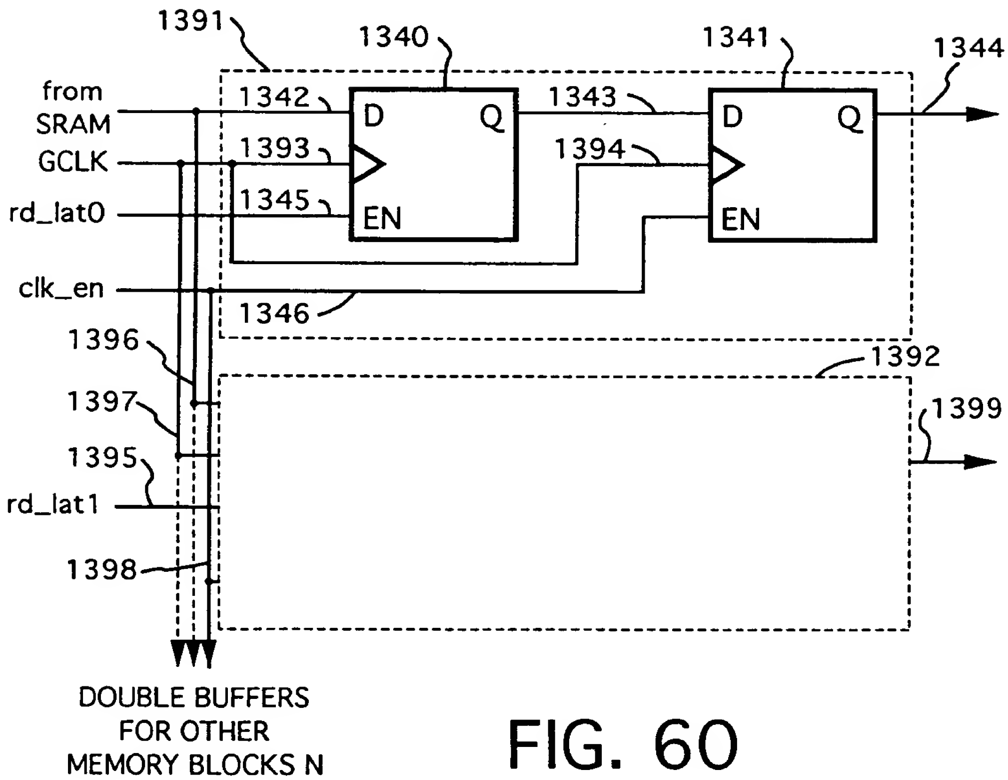


FIG. 60

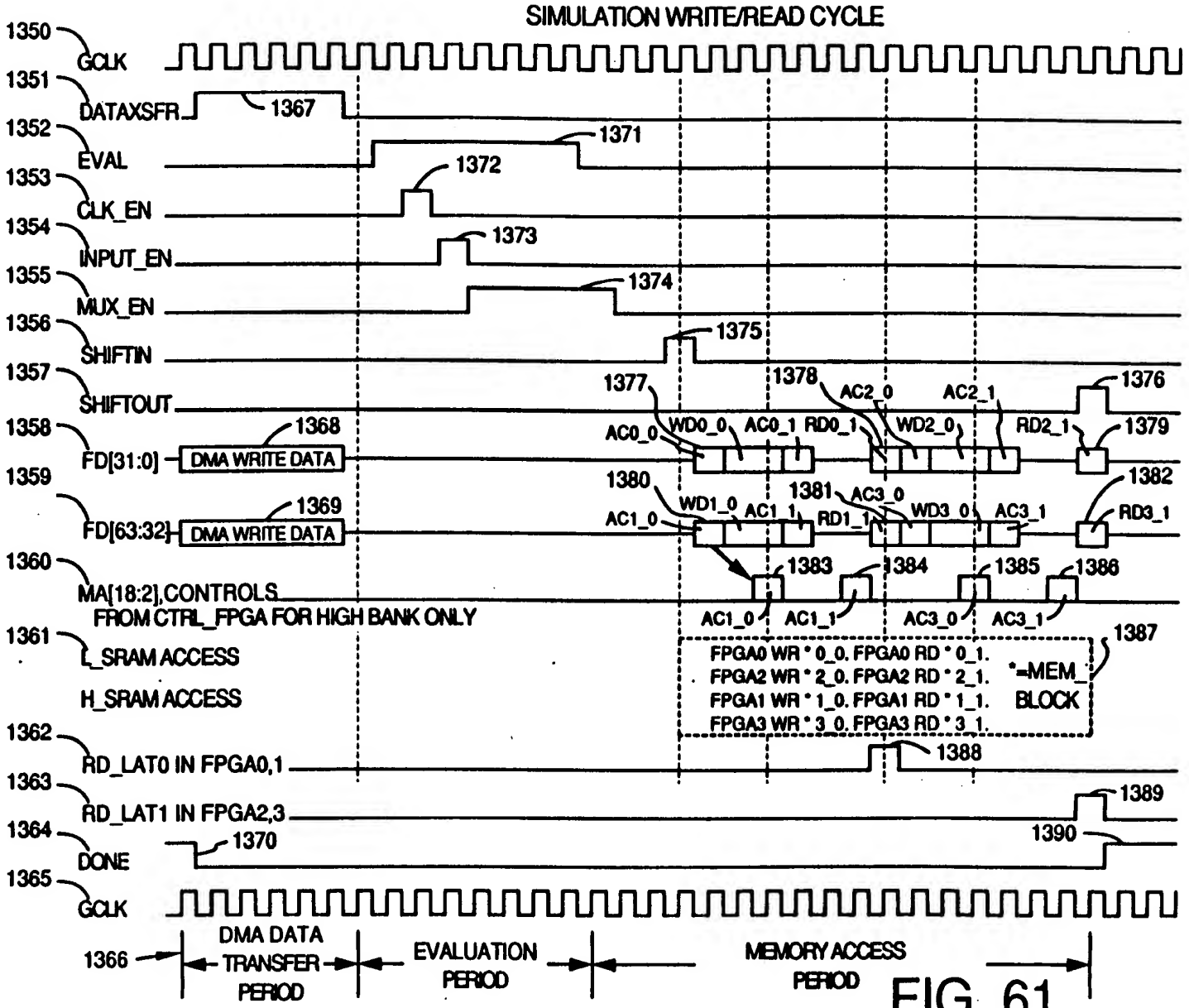

SIMULATION WRITE/READ CYCLE
1350
GCLK
1351
DATAXSFR
1367
1352
EVAL
1371
1353
CLK_EN
1372
1354
INPUT_EN
1373
1355
MUX_EN
1374
1356
SHIFTIN
1375
1357
SHIFTOUT
1358
FD[31:0]
1368
DMA WRITE DATA
1359
FD[63:32]
1369
DMA WRITE DATA
1360
MA[18:2],CONTROLS
FROM CTRL_FPGA FOR HIGH BANK ONLY
1361
L_SRAM ACCESS
H_SRAM ACCESS
1362
RD_LAT0 IN FPGA0,1
1363
RD_LAT1 IN FPGA2,3
1364
DONE
1370
1365
GCLK
1377
1378
1376
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
AC0_0
WD0_0
AC0_1
RD0_1
AC2_0
WD2_0
AC2_1
RD2_1
AC1_0
WD1_0
AC1_1
RD1_1
AC3_0
WD3_0
AC3_1
RD3_1
AC1_0
AC1_1
AC3_0
AC3_1
FPGA0 WR * 0_0. FPGA0 RD * 0_1.
FPGA2 WR * 2_0. FPGA2 RD * 2_1.
FPGA1 WR * 1_0. FPGA1 RD * 1_1.
FPGA3 WR * 3_0. FPGA3 RD * 3_1.
*=MEM_
BLOCK
1366
DMA DATA TRANSFER PERIOD
EVALUATION PERIOD
MEMORY ACCESS PERIOD

FIG. 61

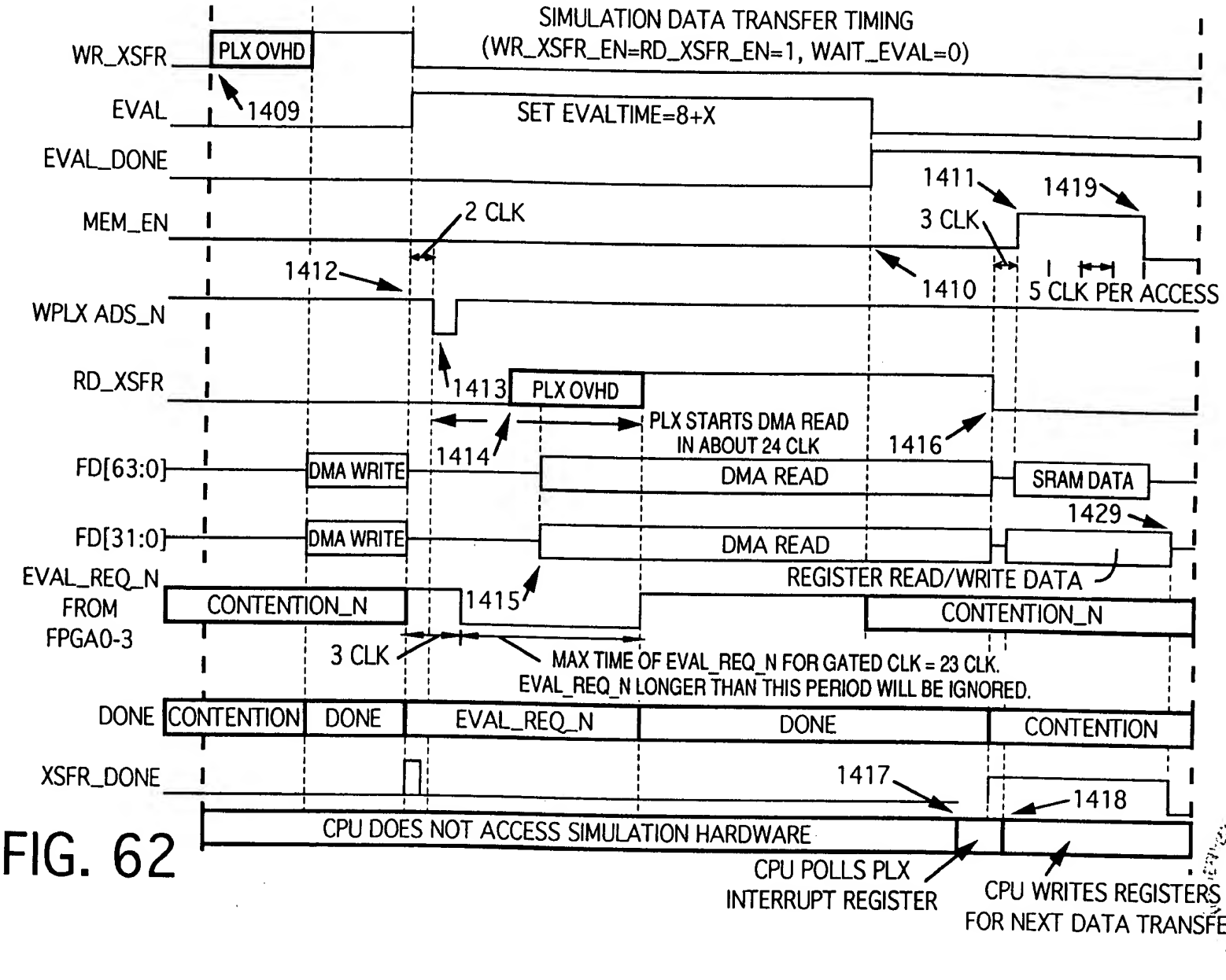

SIMULATION DATA TRANSFER TIMING
(WR_XSFR_EN=RD_XSFR_EN=1, WAIT_EVAL=0)
WR_XSFR
PLX OVHD
EVAL
1409
SET EVALTIME=8+X
EVAL_DONE
1411
1419
MEM_EN
2 CLK
3 CLK
1412
1410
5 CLK PER ACCESS
WPLX ADS_N
RD_XSFR
1413
PLX OVHD
PLX STARTS DMA READ
IN ABOUT 24 CLK
1414
1416
FD[63:0]
DMA WRITE
DMA READ
SRAM DATA
1429
FD[31:0]
DMA WRITE
DMA READ
REGISTER READ/WRITE DATA
EVAL_REQ_N
FROM
FPGA0-3
CONTENTION_N
1415
CONTENTION_N
3 CLK
MAX TIME OF EVAL_REQ_N FOR GATED CLK = 23 CLK.
EVAL_REQ_N LONGER THAN THIS PERIOD WILL BE IGNORED.
DONE
CONTENTION
DONE
EVAL_REQ_N
DONE
CONTENTION
XSFR_DONE
1417
1418
CPU DOES NOT ACCESS SIMULATION HARDWARE
CPU POLLS PLX
INTERRUPT REGISTER
CPU WRITES REGISTERS
FOR NEXT DATA TRANSFER

FIG. 62

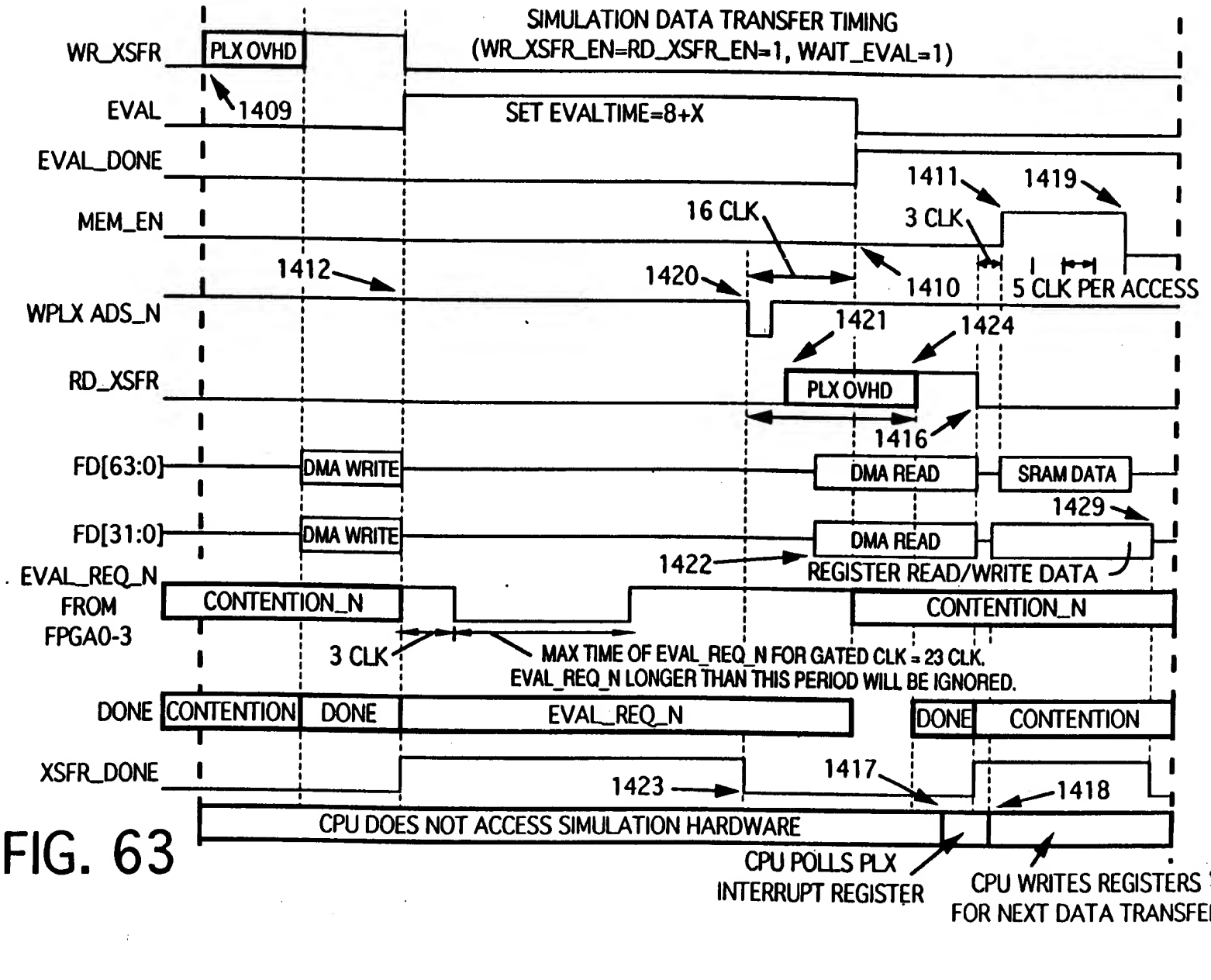

SIMULATION DATA TRANSFER TIMING
(WR_XSFR_EN=RD_XSFR_EN=1, WAIT_EVAL=1)
WR_XSFR
PLX OVHD
EVAL
1409
SET EVALTIME=8+X
EVAL_DONE
1411
1419
MEM_EN
16 CLK
3 CLK
1412
1420
1410
5 CLK PER ACCESS
WPLX ADS_N
1421
1424
RD_XSFR
PLX OVHD
1416
FD[63:0]
DMA WRITE
DMA READ
SRAM DATA
1429
FD[31:0]
DMA WRITE
DMA READ
1422
REGISTER READ/WRITE DATA
EVAL_REQ_N FROM FPGA0-3
CONTENTION_N
CONTENTION_N
3 CLK
MAX TIME OF EVAL_REQ_N FOR GATED CLK = 23 CLK.
EVAL_REQ_N LONGER THAN THIS PERIOD WILL BE IGNORED.
DONE
CONTENTION
DONE
EVAL_REQ_N
DONE
CONTENTION
XSFR_DONE
1423
1417
1418
CPU DOES NOT ACCESS SIMULATION HARDWARE
CPU POLLS PLX INTERRUPT REGISTER
CPU WRITES REGISTERS FOR NEXT DATA TRANSFER

FIG. 63

Typical User Design of PCI Add-on Cards

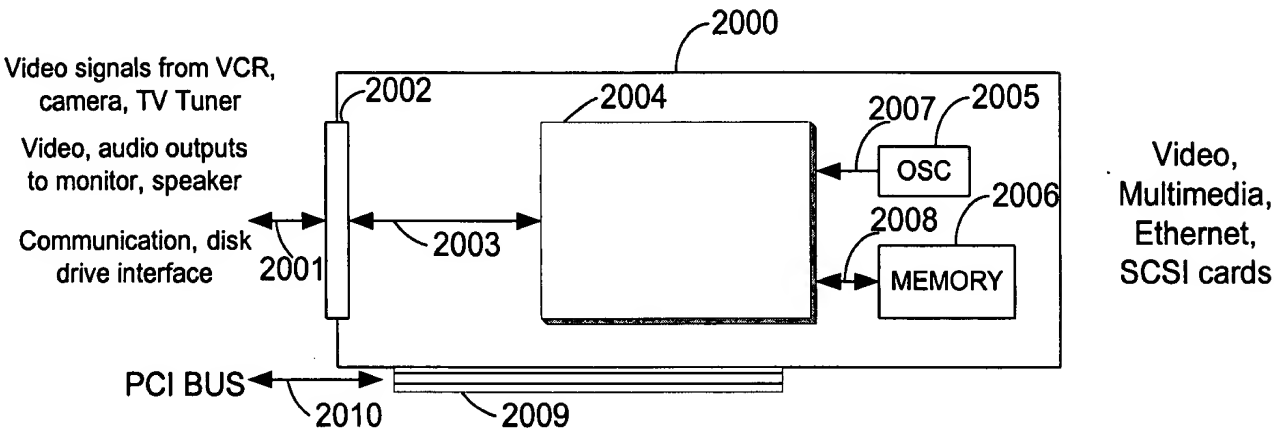
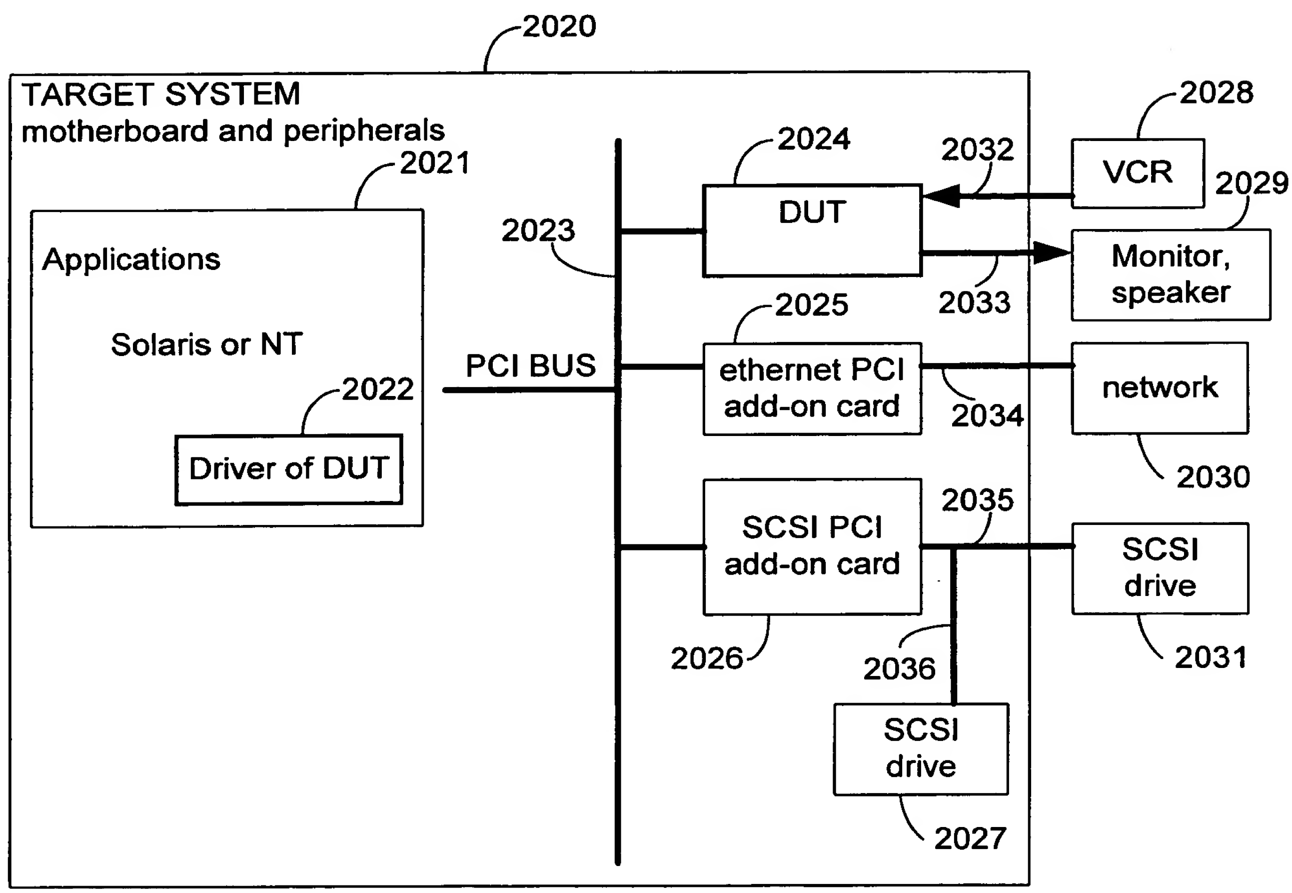


FIG. 64

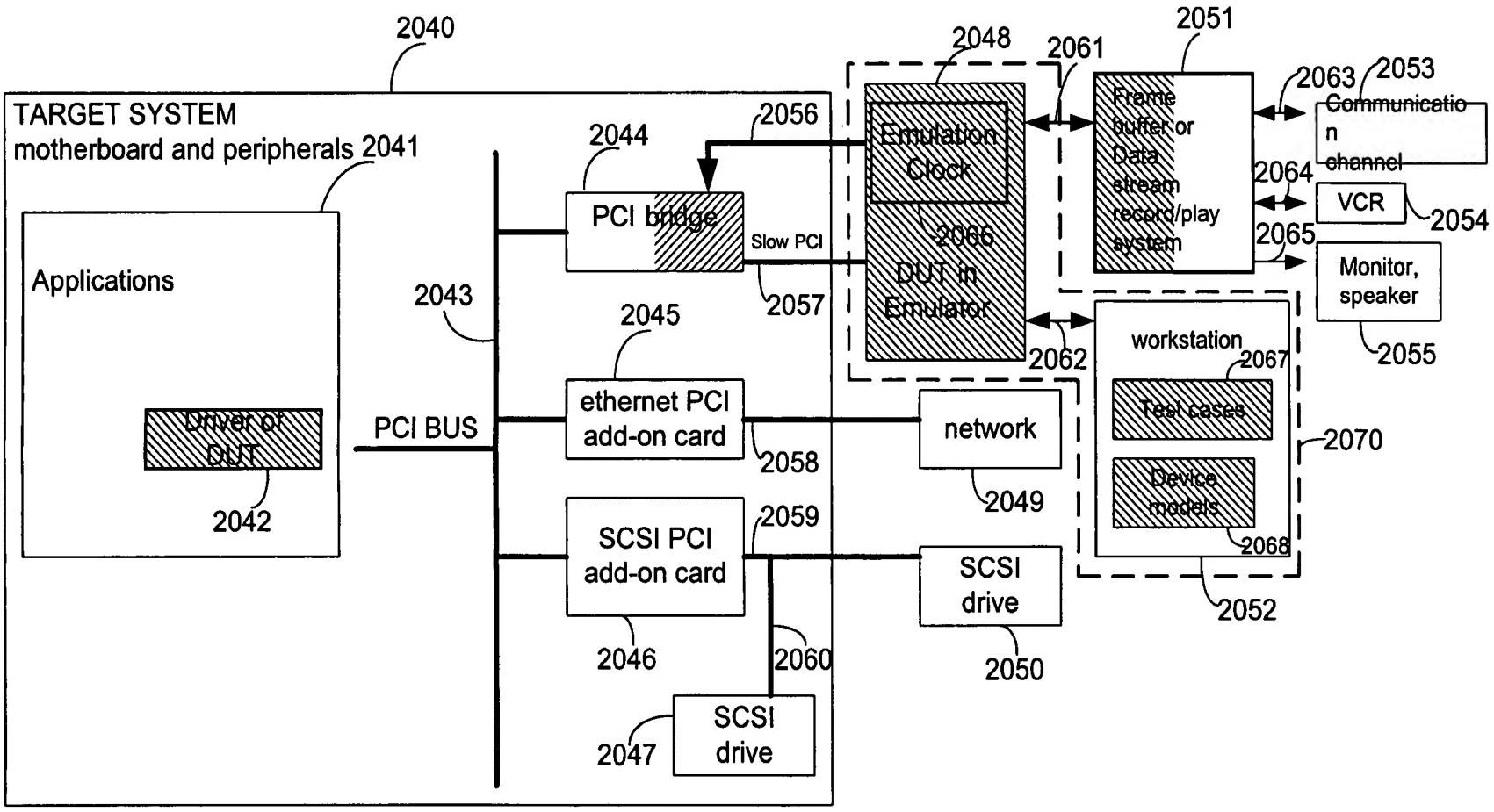
Typical Hardware/Software Co-Verification



: DUT (Device Under Test)

FIG. 65

Typical Co-Verification by Using Emulator



: running time at emulation speed

The rest of the target system is running at full speed.

FIG. 66

SIMULATION

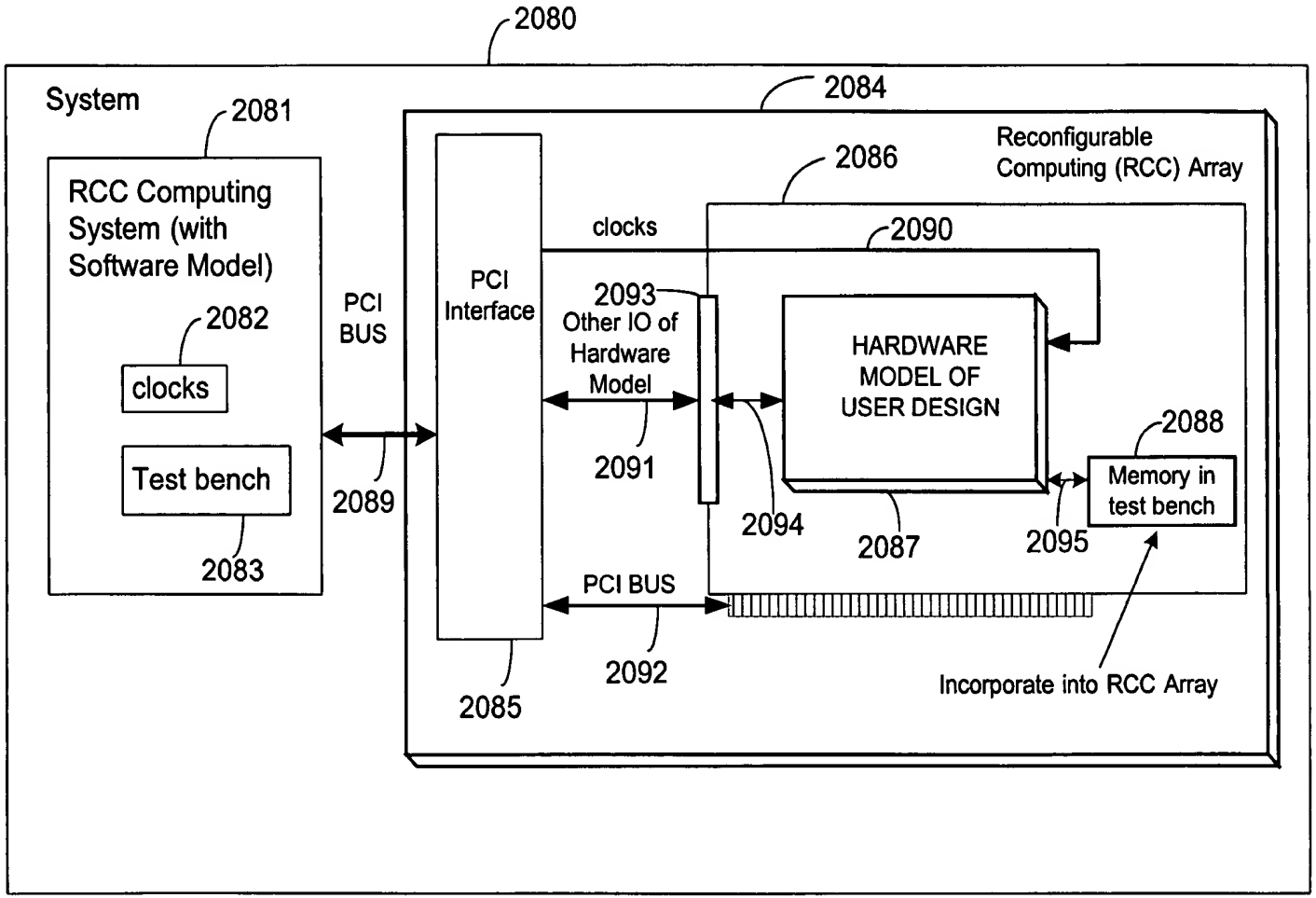


FIG. 67

CO-VERIFICATION WITHOUT EXTERNAL I/O

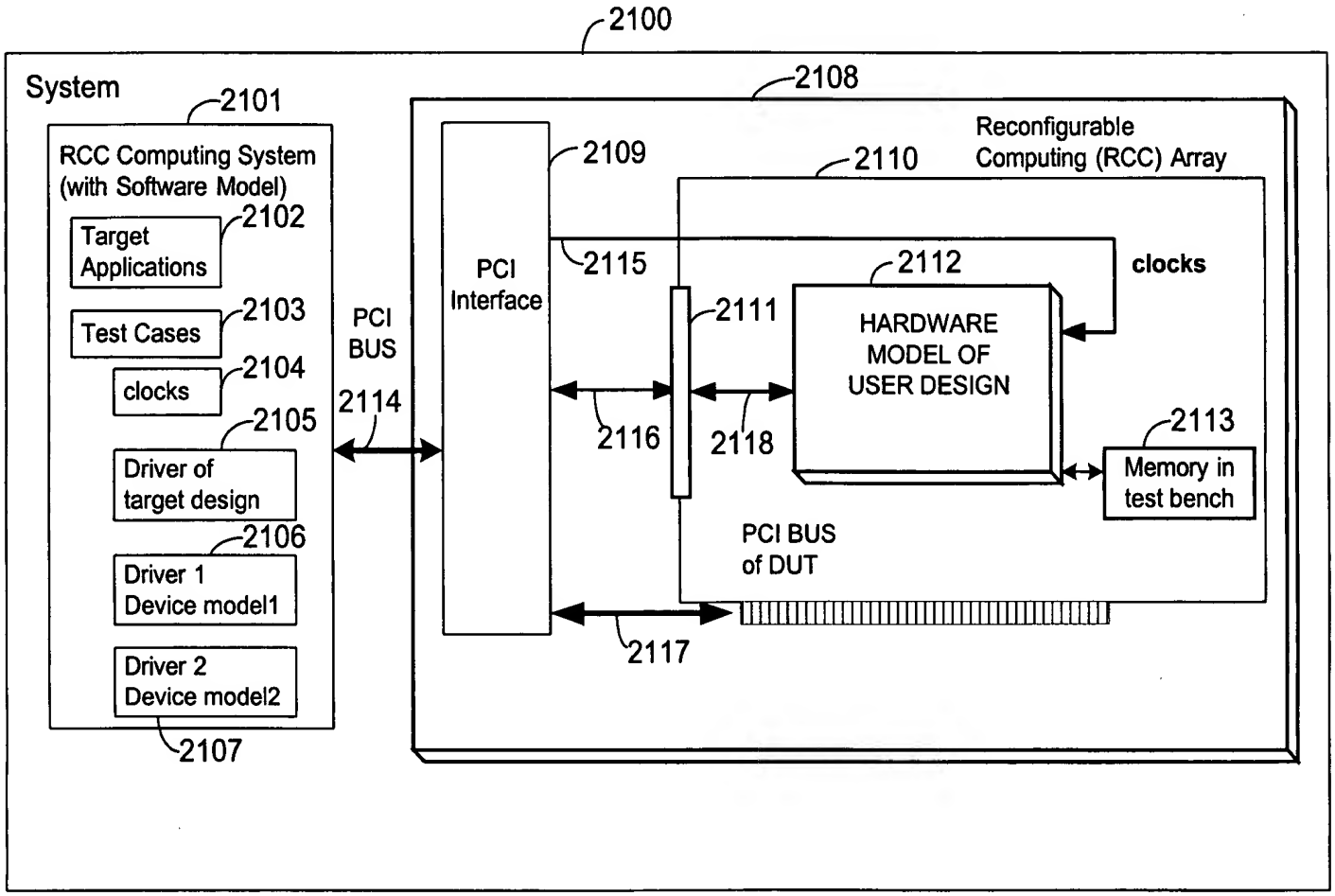


FIG. 68

CO-VERIFICATION WITH EXTERNAL I/O

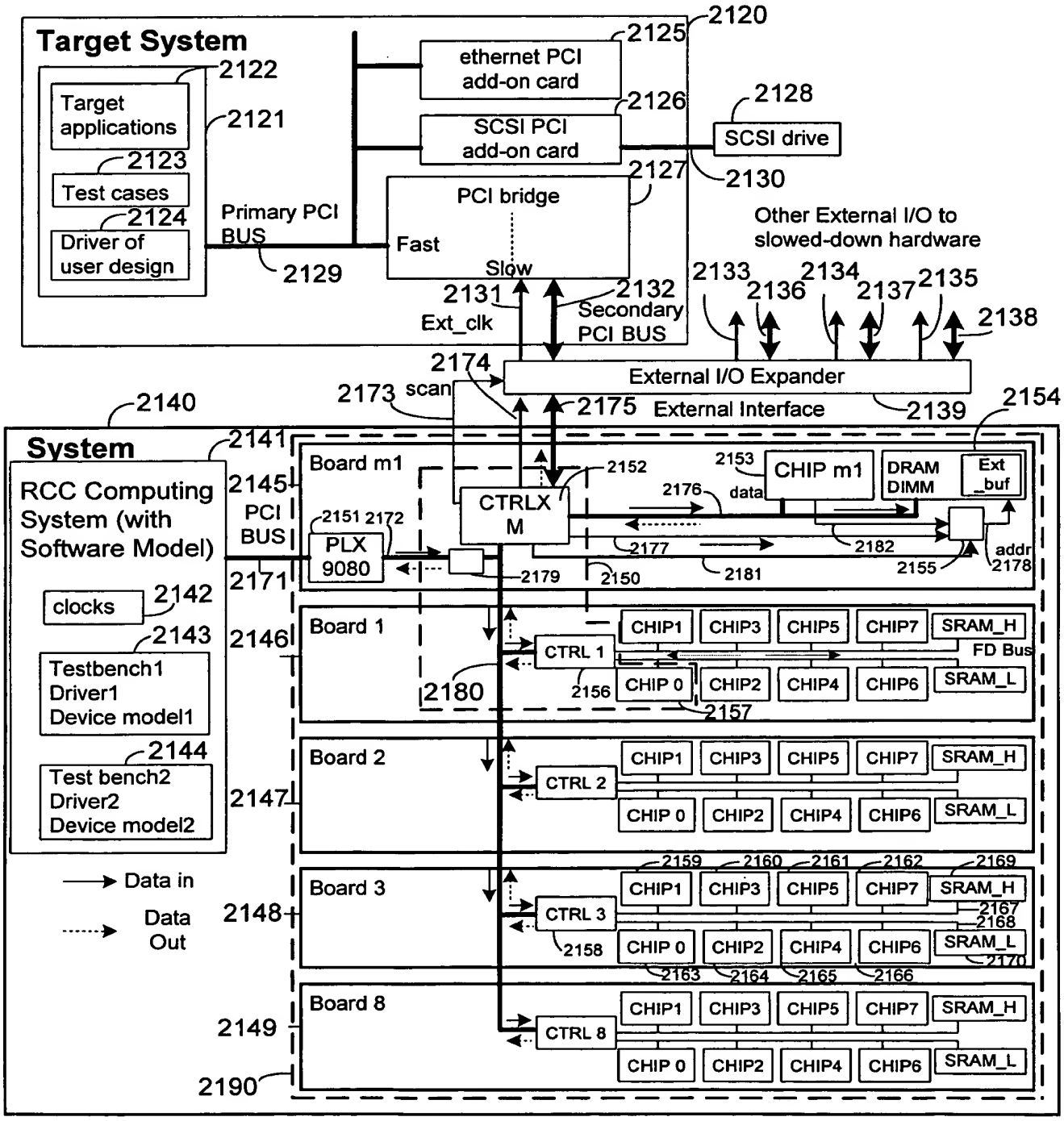


FIG. 69

CONTROL OF DATA-IN CYCLE

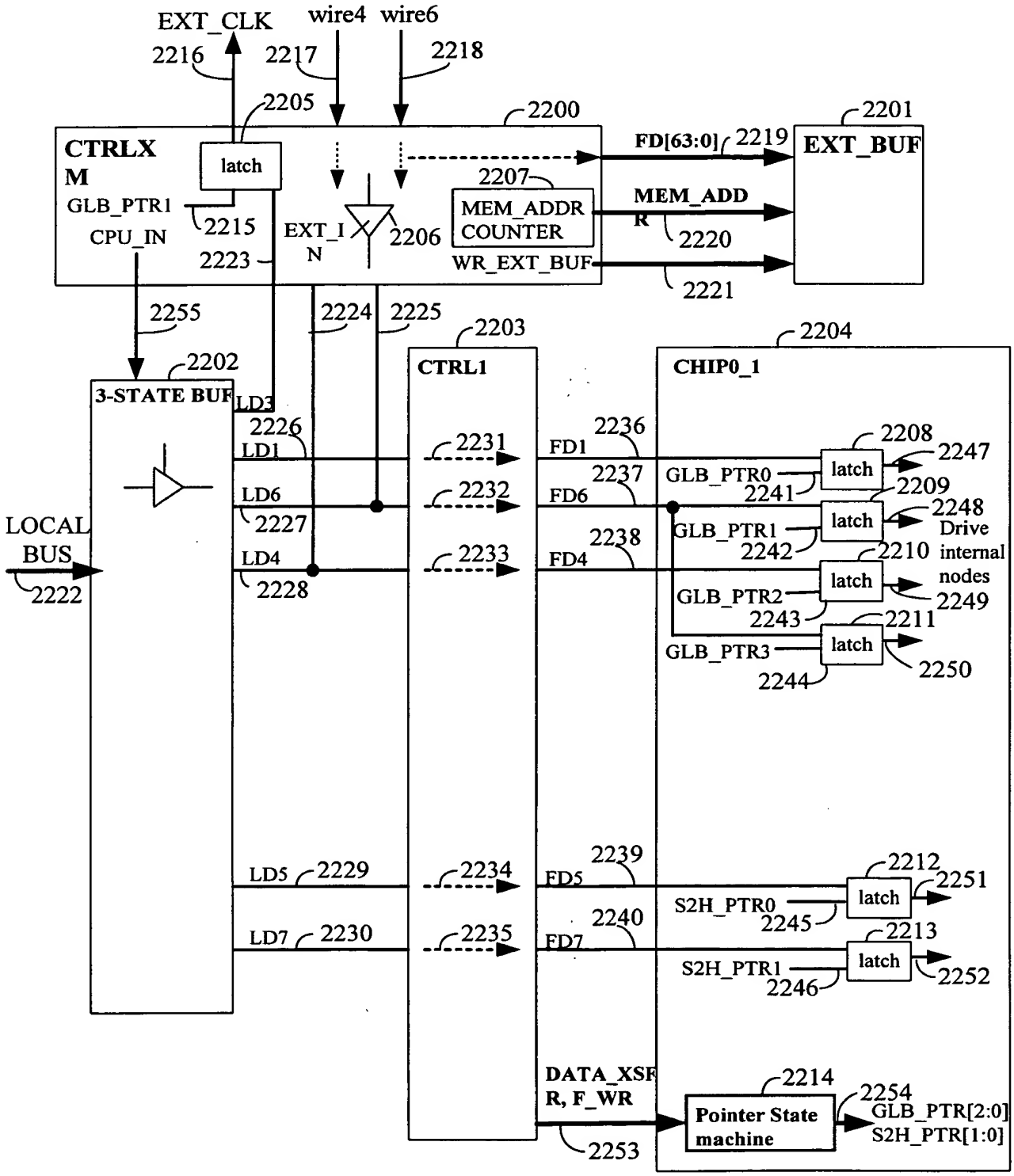


FIG. 70

CONTROL OF DATA-OUT CYCLE

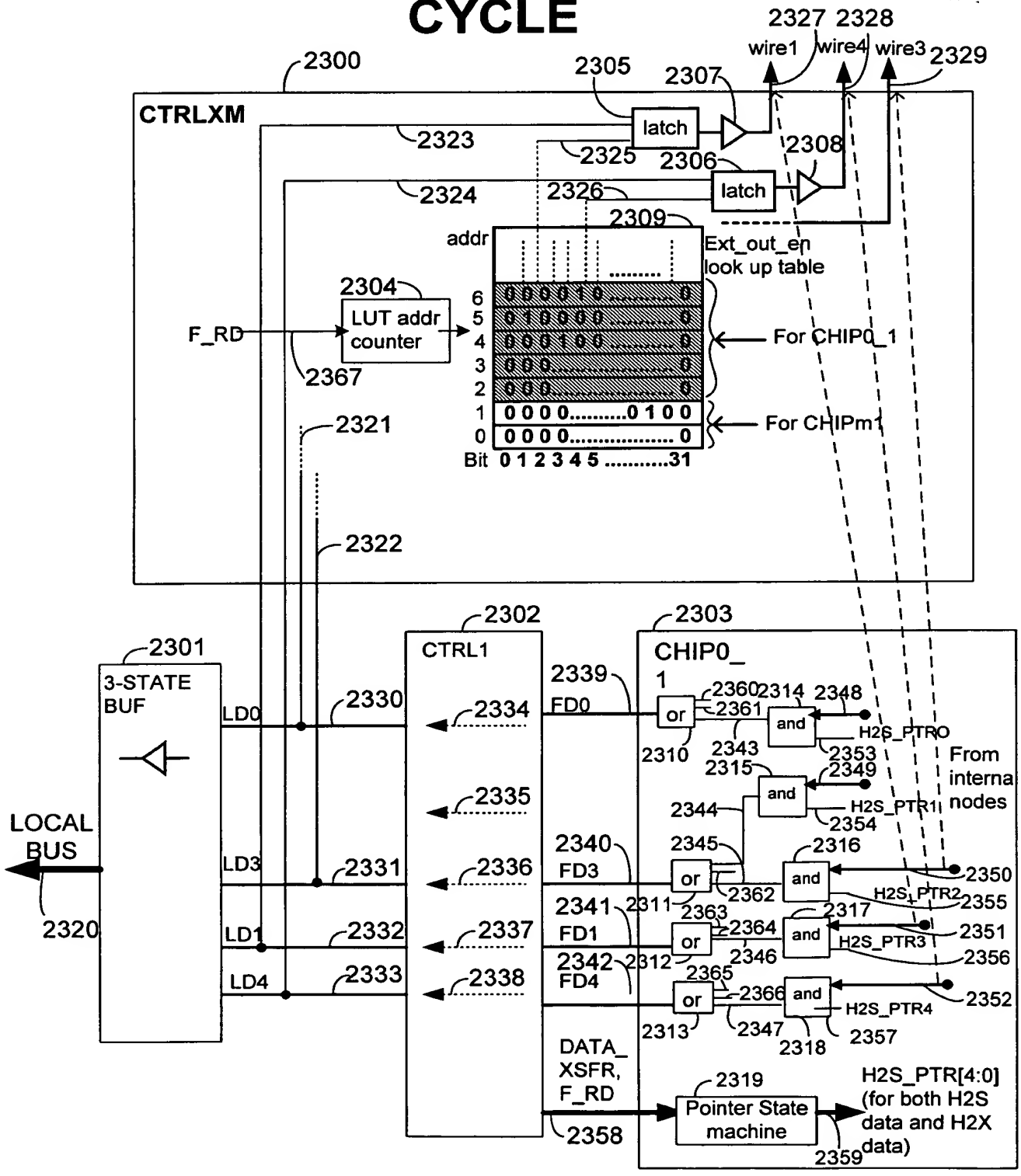


FIG. 71

CONTROL OF DATA-IN CYCLE

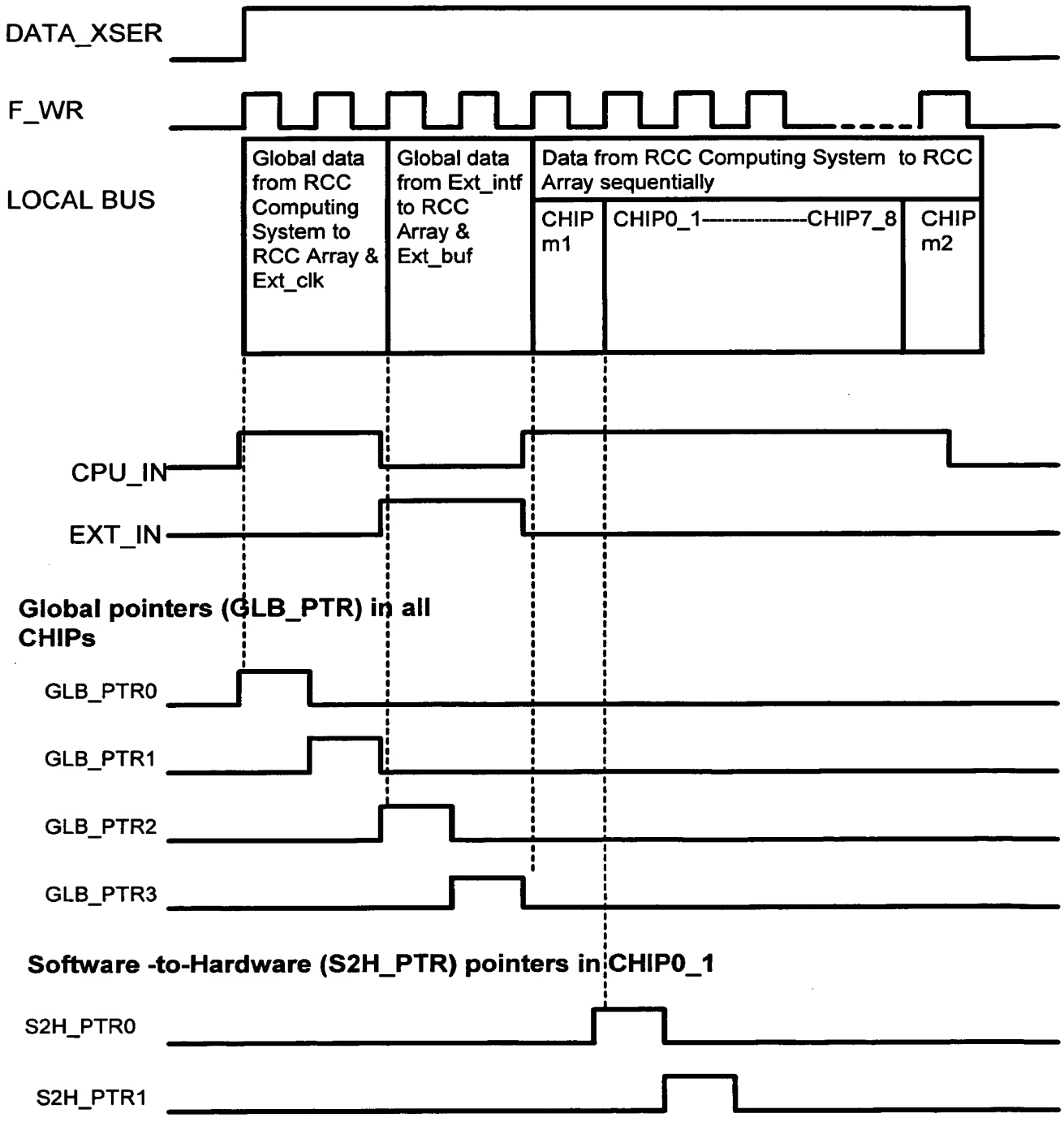


FIG. 72

CONTROL OF DATA-OUT CYCLE

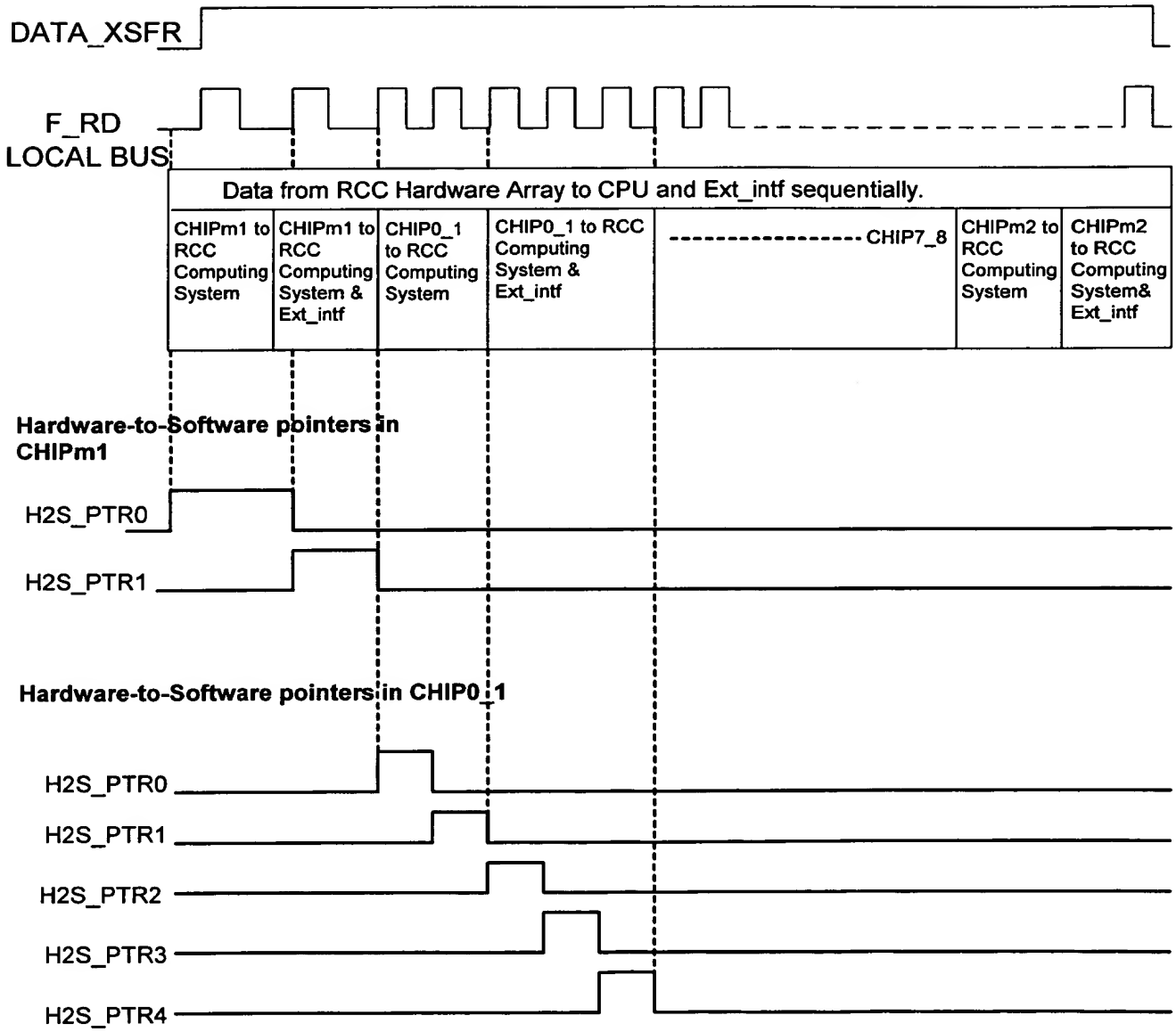


FIG. 73

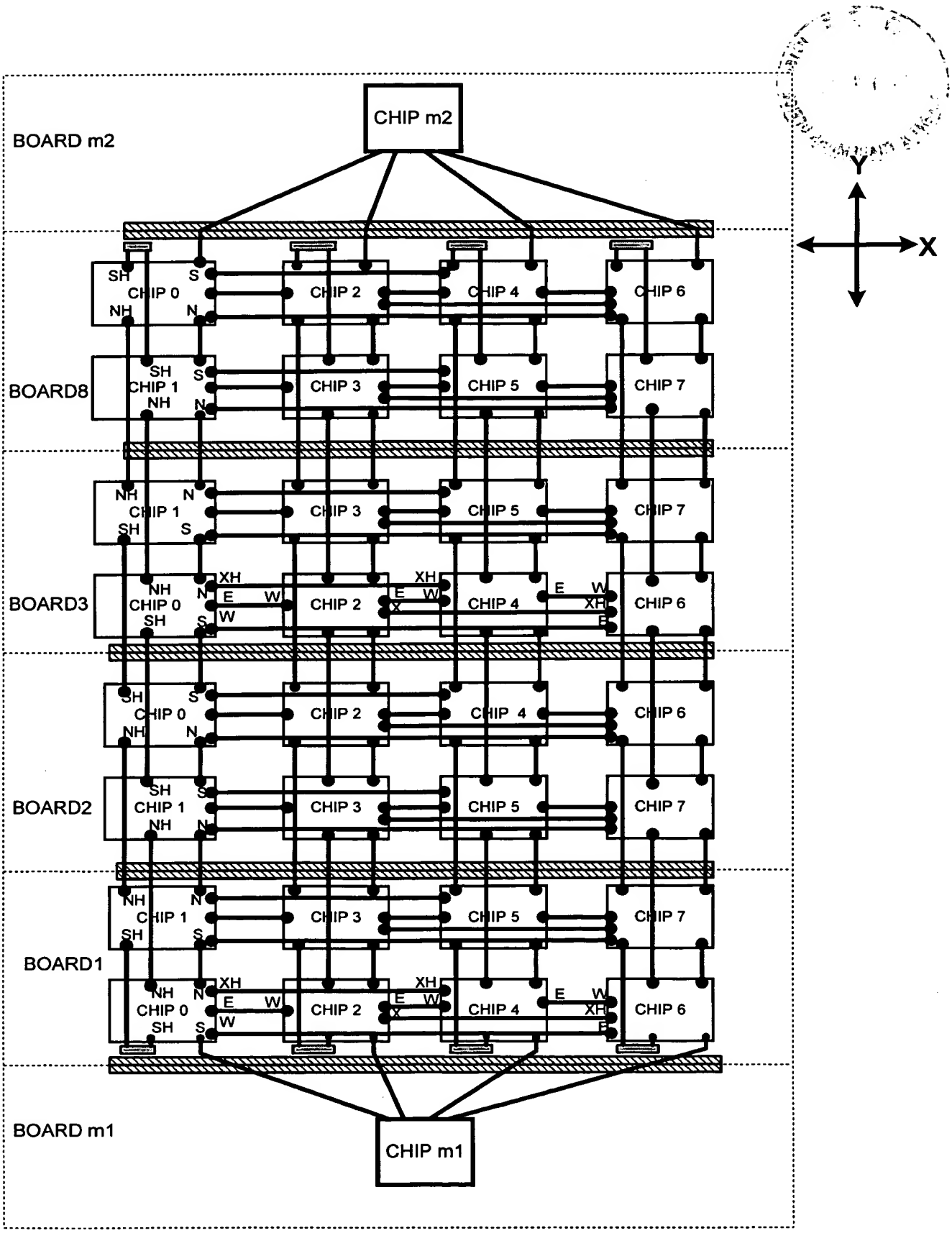


FIG. 74

SHIFT REGISTER

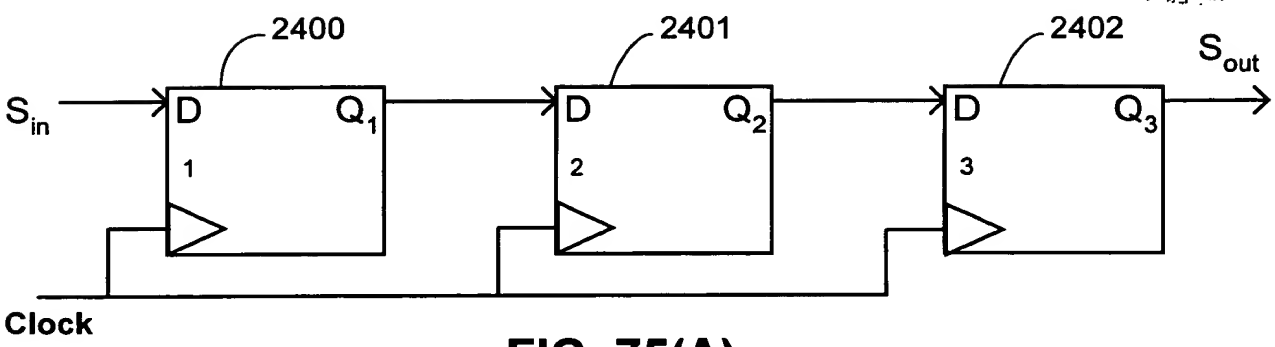


FIG. 75(A)

HOLD TIME ASSUMPTION FOR SHIFT REGISTER

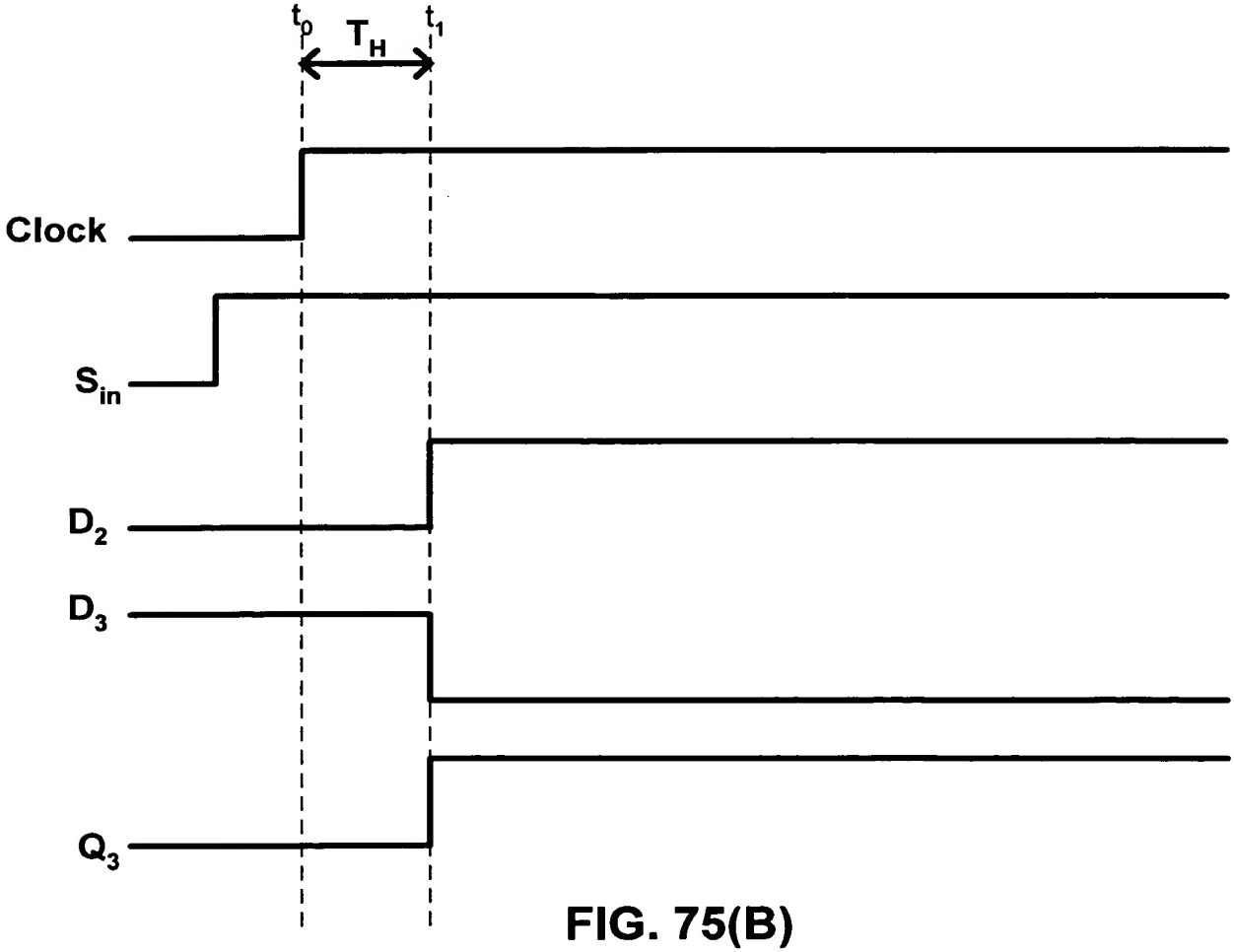


FIG. 75(B)

MULTIPLE FPGA MAPPING FOR SHIFT REGISTER

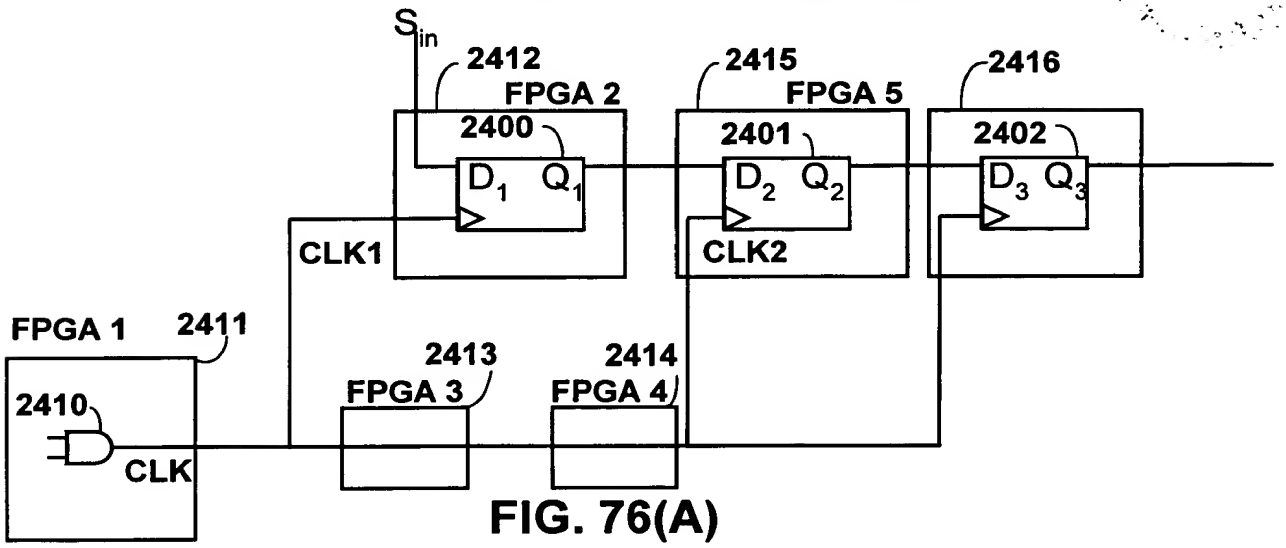


FIG. 76(A)

HOLD TIME VIOLATION BY LONG CLOCK SKEW

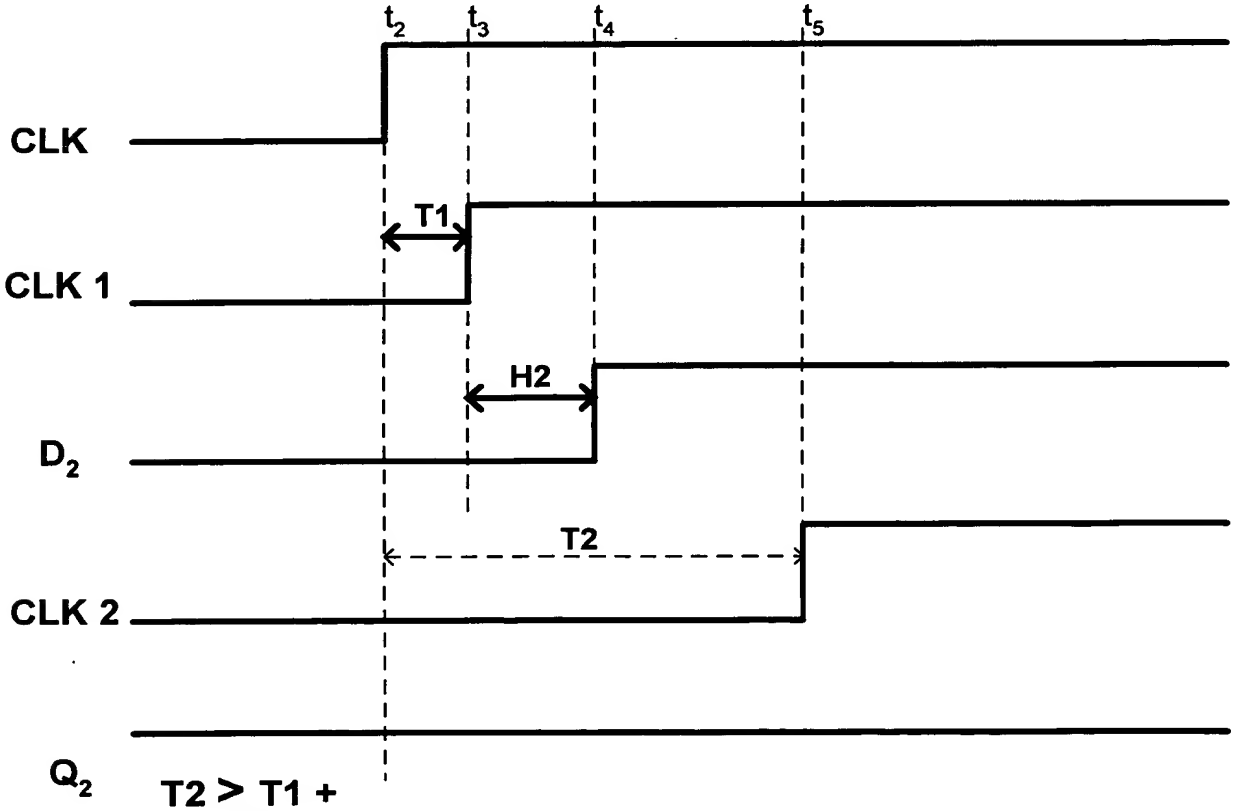


FIG. 76(B)

CLOCK GLITCH PROBLEM

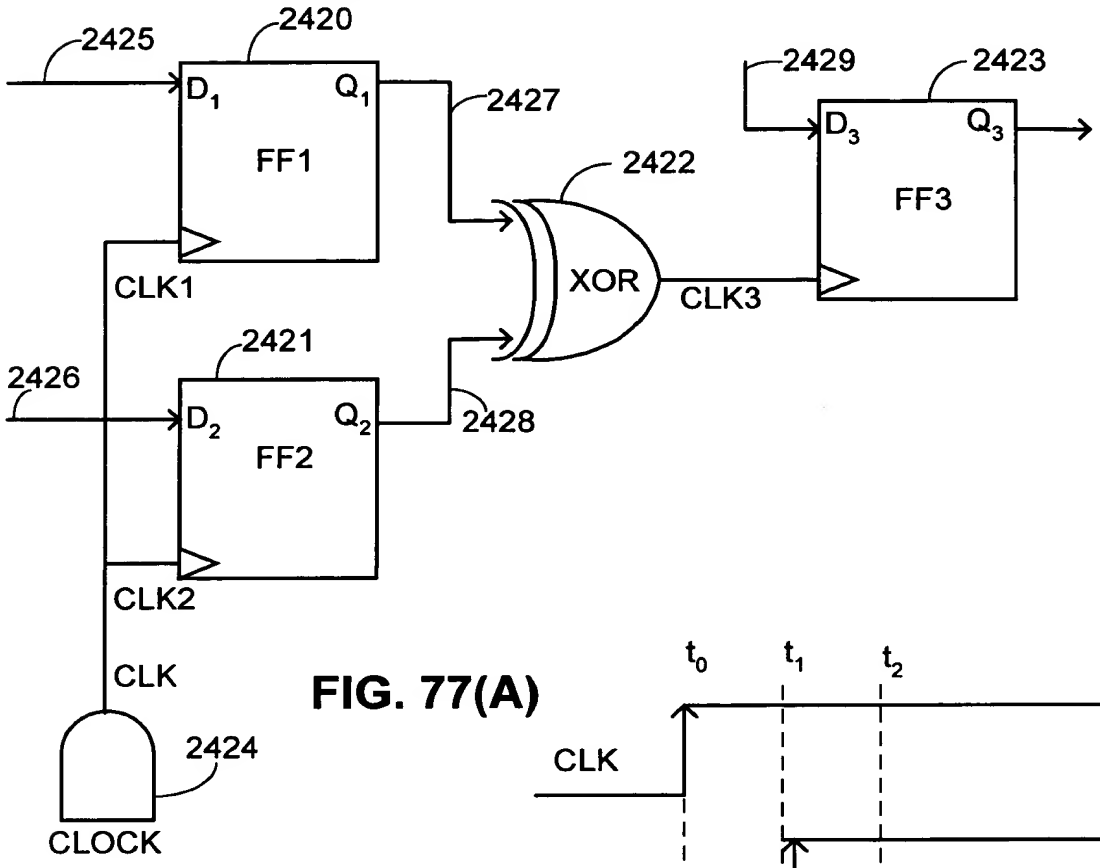


FIG. 77(A)

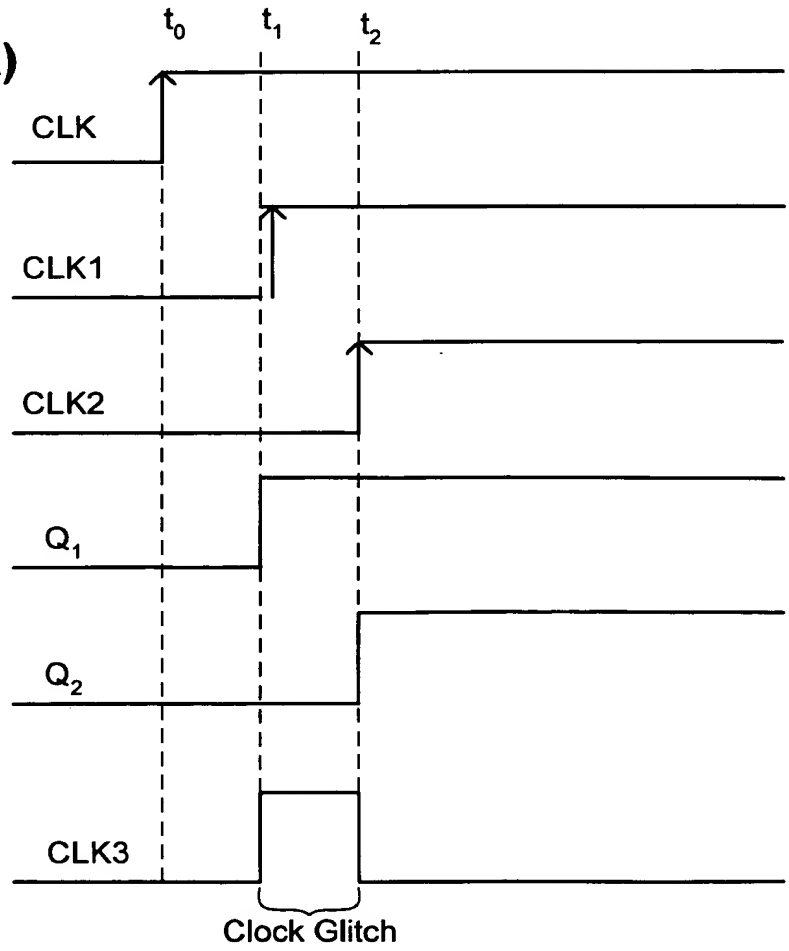
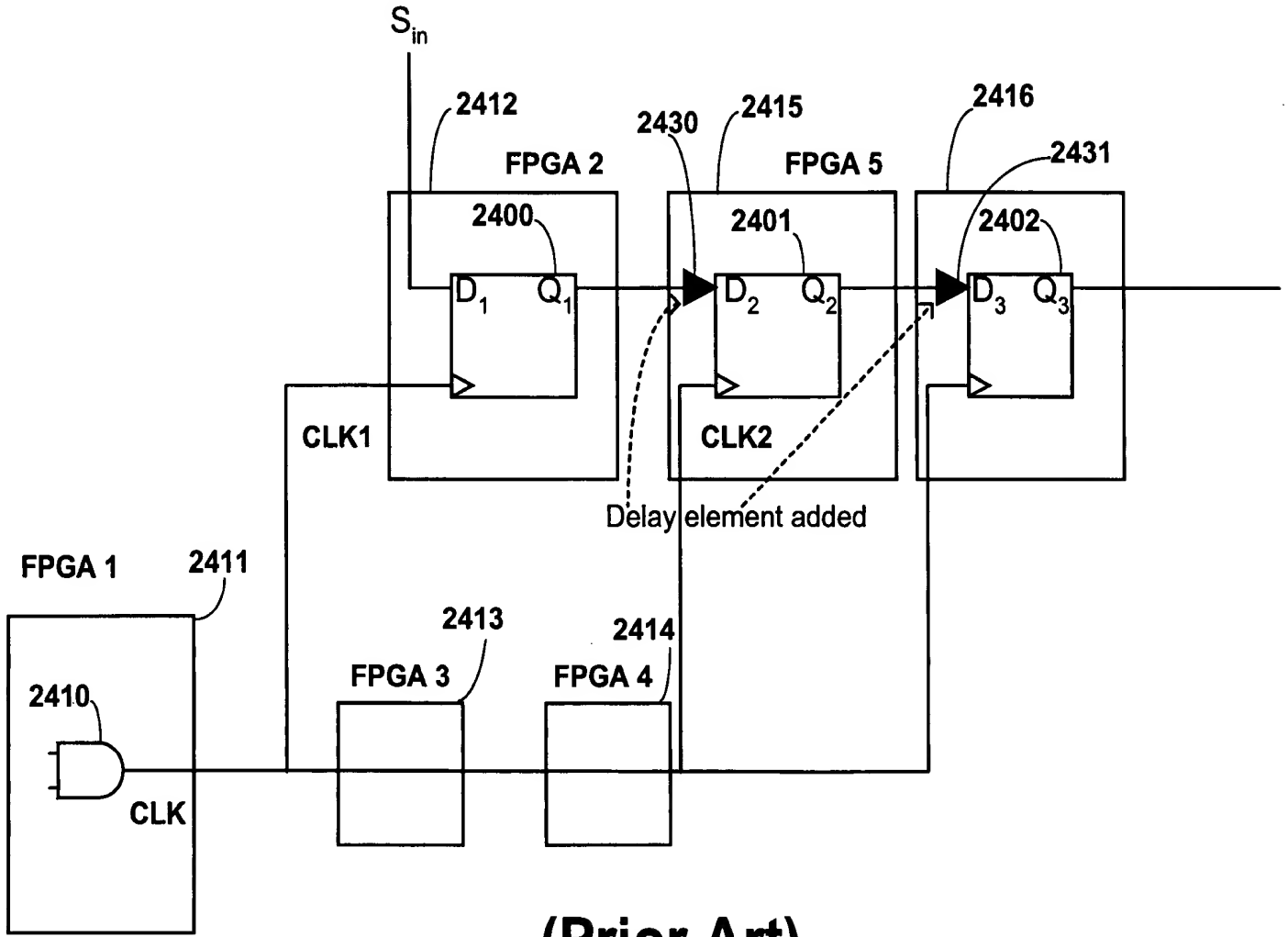


FIG. 77(B)

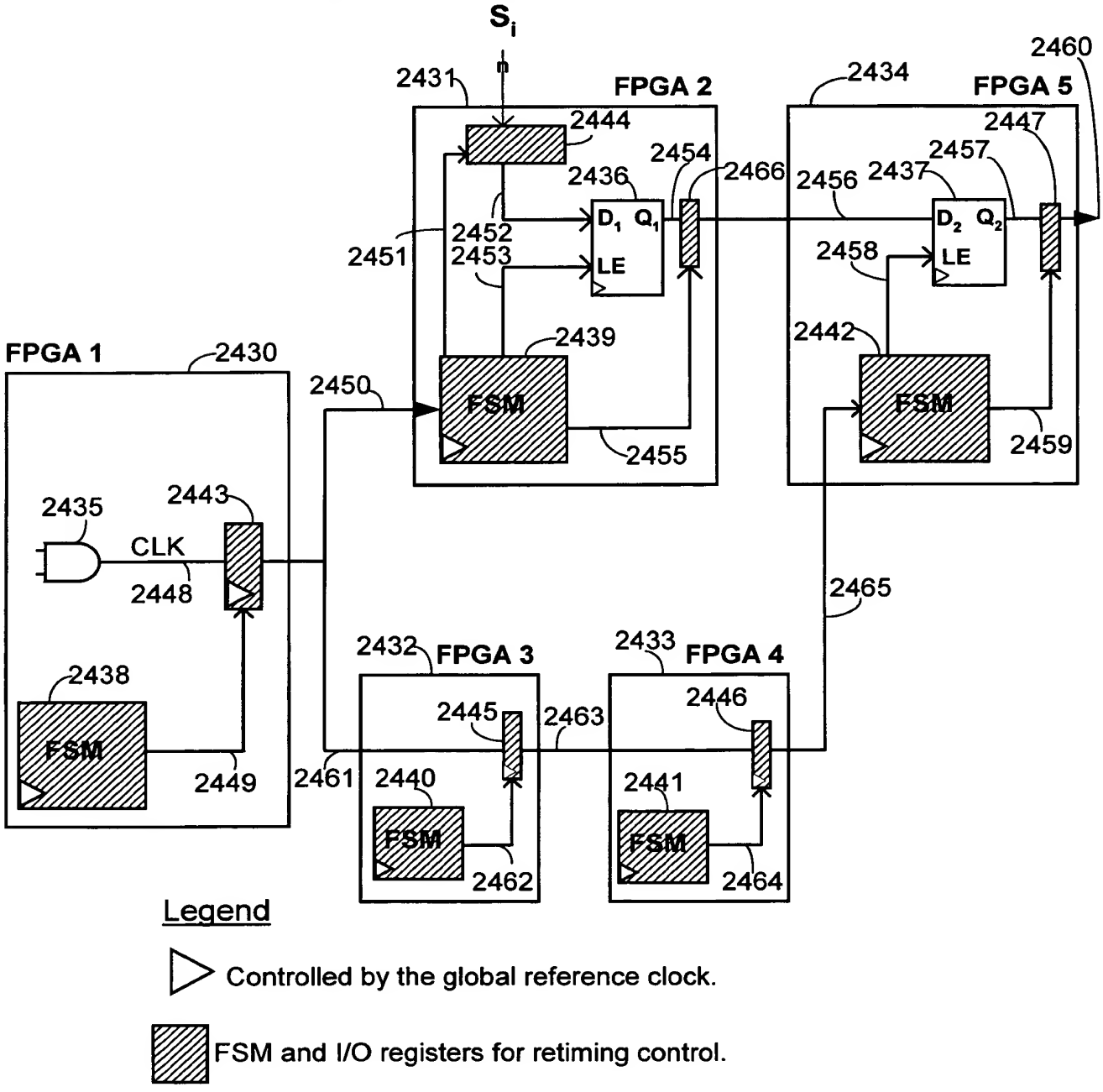
TIMING ADJUSTMENT BY ADDING DELAY



(Prior Art)

FIG. 78

GLOBAL RETIMING



(Prior Art)

FIG. 79

TIGF LATCH

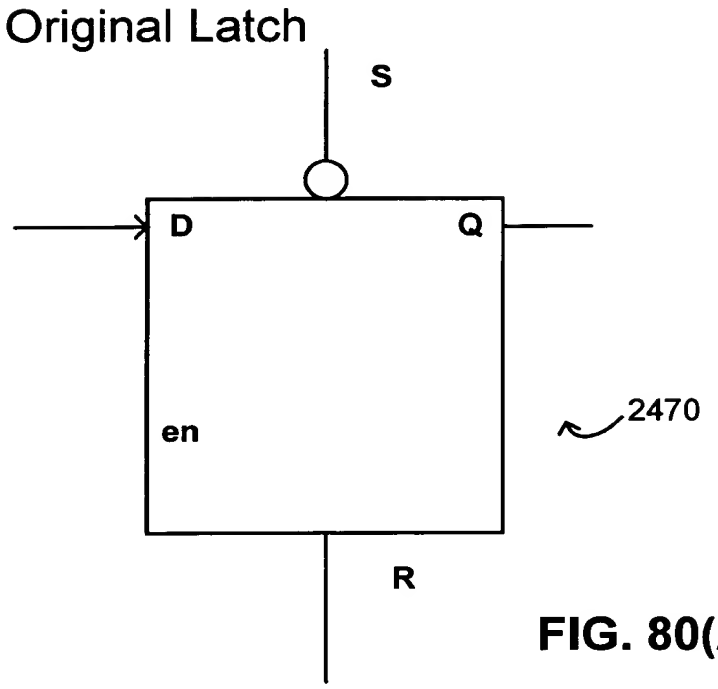


FIG. 80(A)

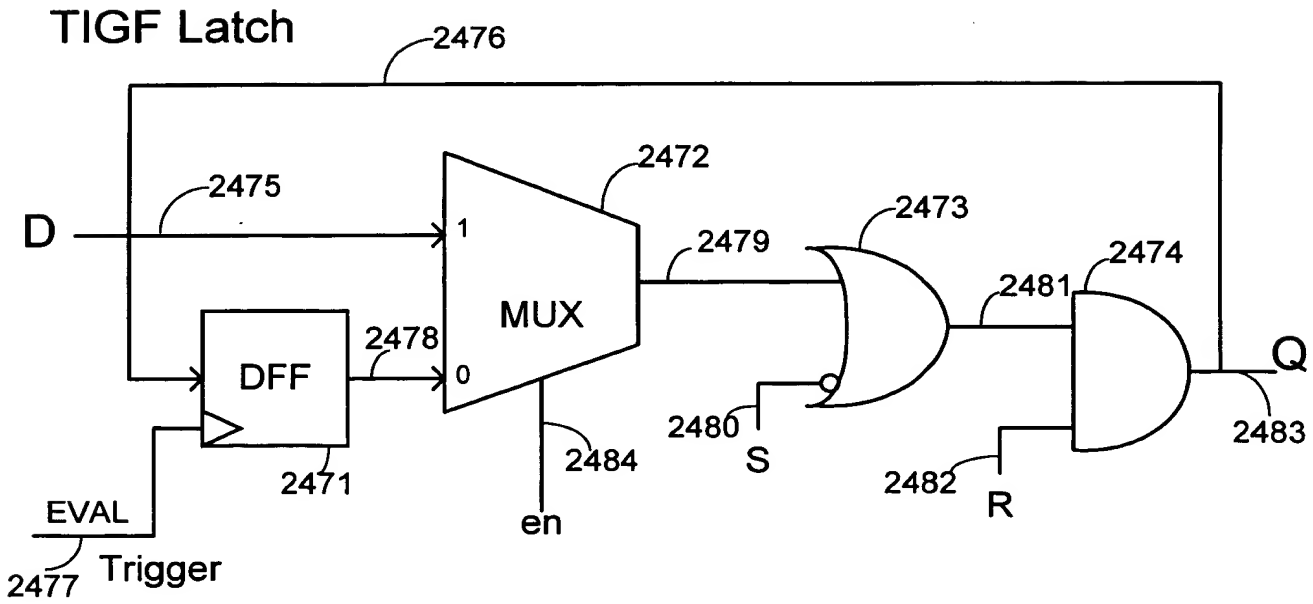


FIG. 80(B)

TIGF DFF

Original DFF

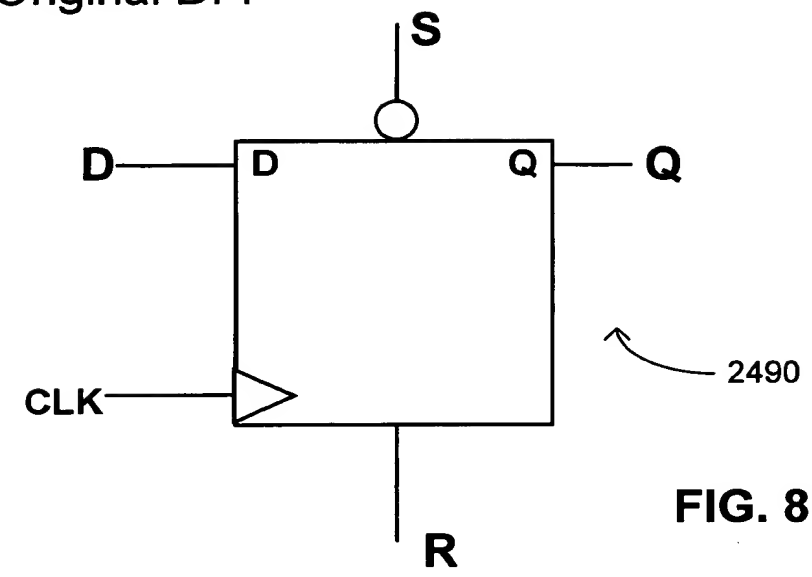


FIG. 81(A)

TIGF DFF and Edge Detector

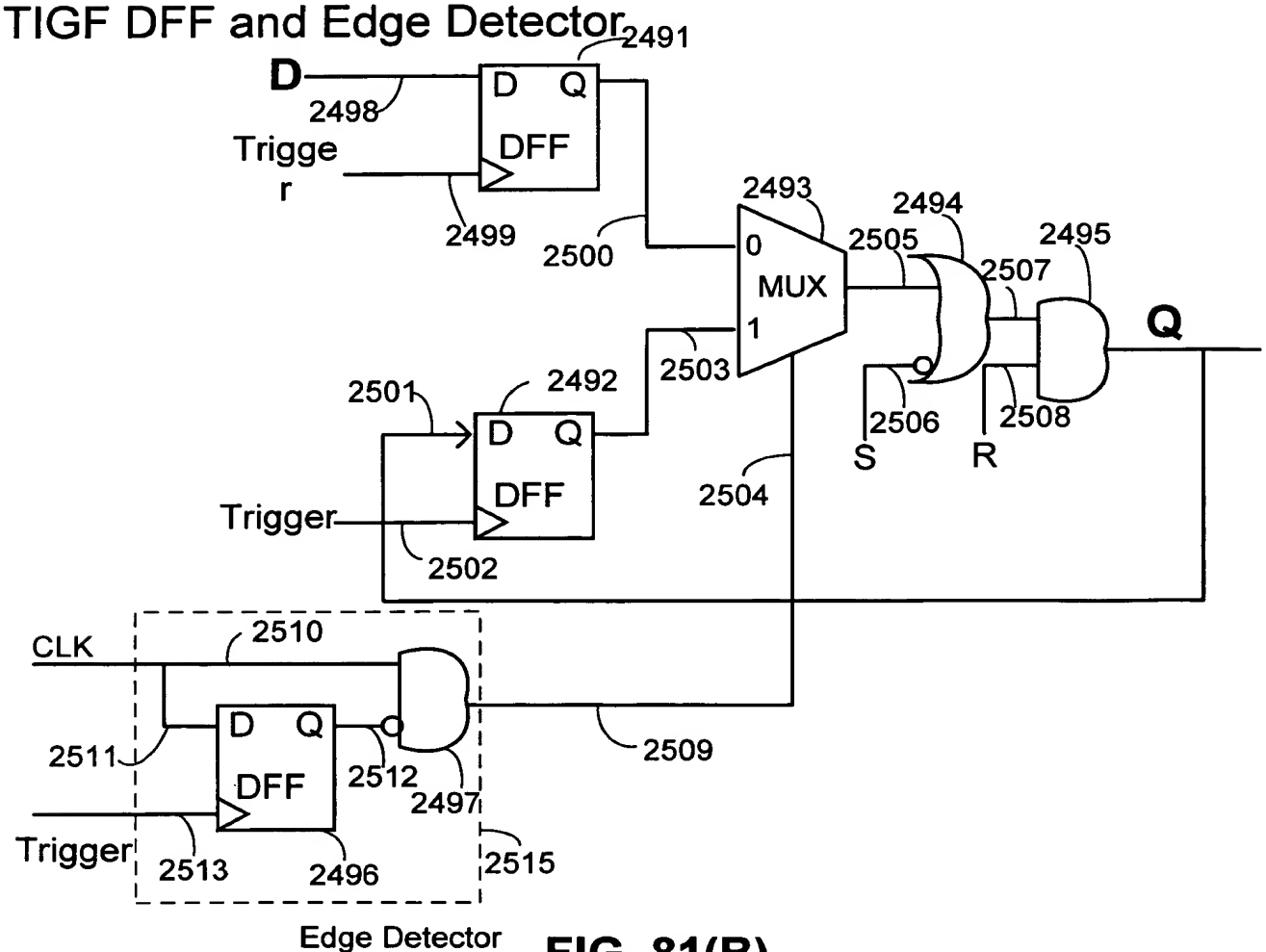


FIG. 81(B)

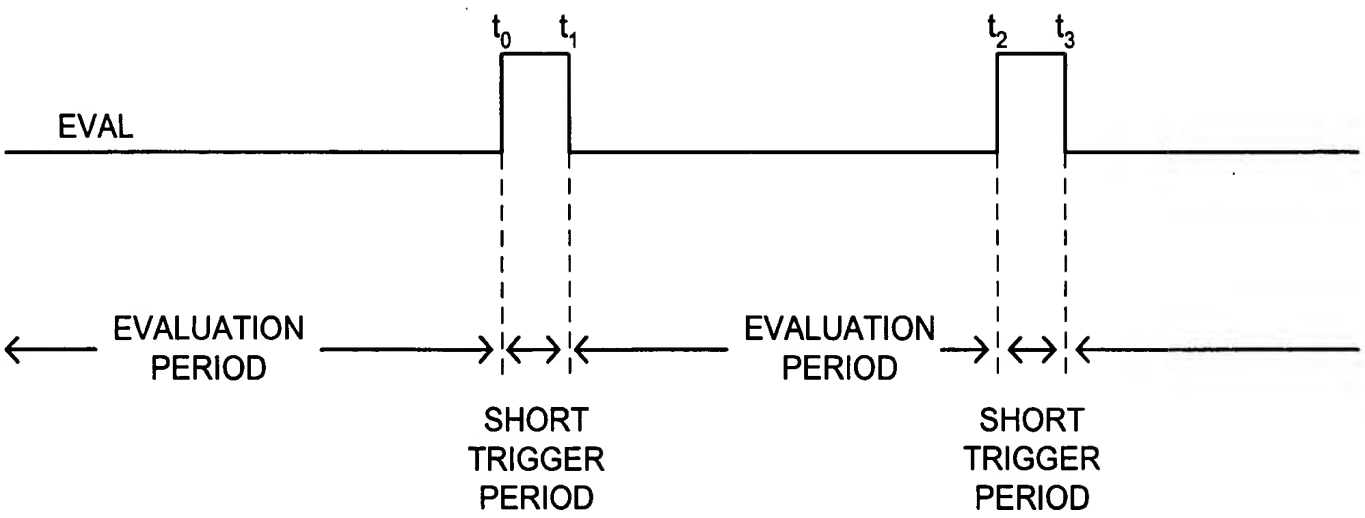

GLOBAL TRIGGER SIGNAL
t_0
t_1
t_2
t_3
EVAL
EVALUATION PERIOD
EVALUATION PERIOD
SHORT TRIGGER PERIOD
SHORT TRIGGER PERIOD

FIG. 82

RCC System

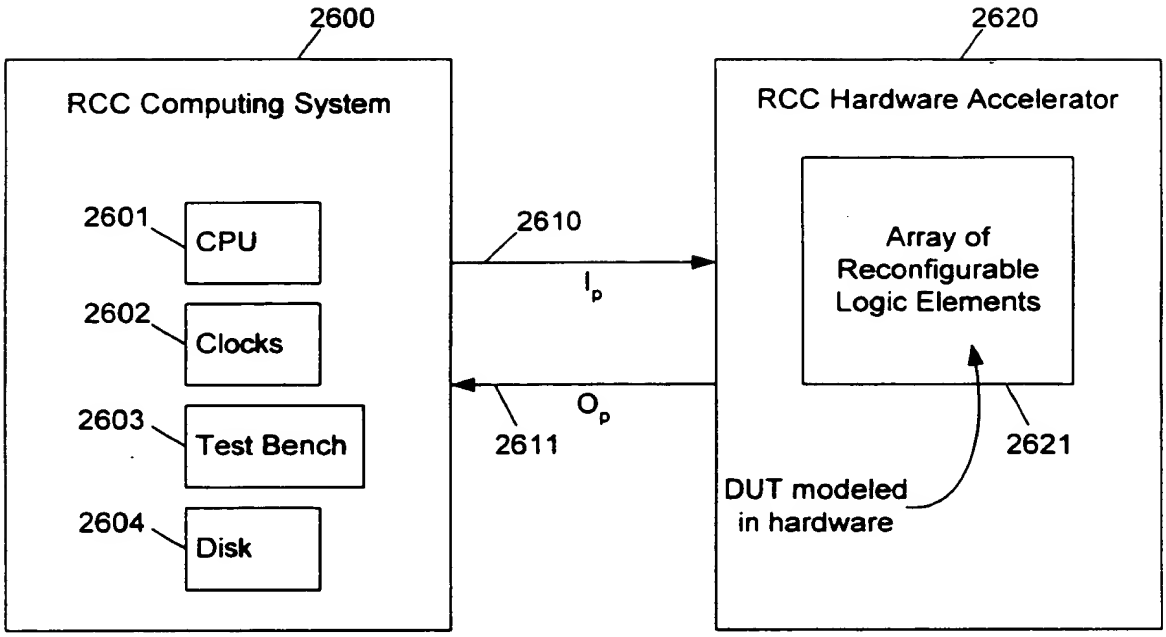


FIG. 83

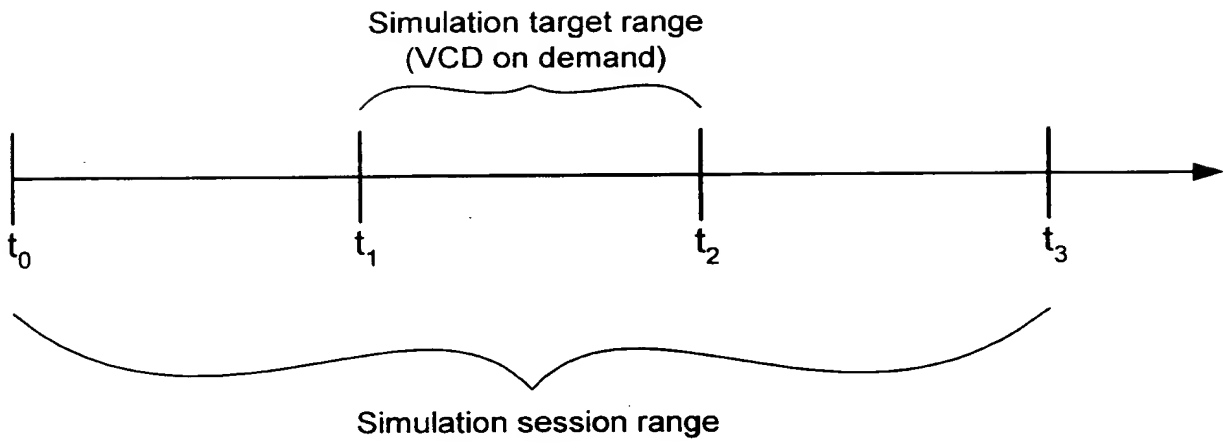

Simulation target range
(VCD on demand)
t_0
t_1
t_2
t_3
Simulation session range

FIG. 84

SINGLE-ROW FPGA PER BOARD

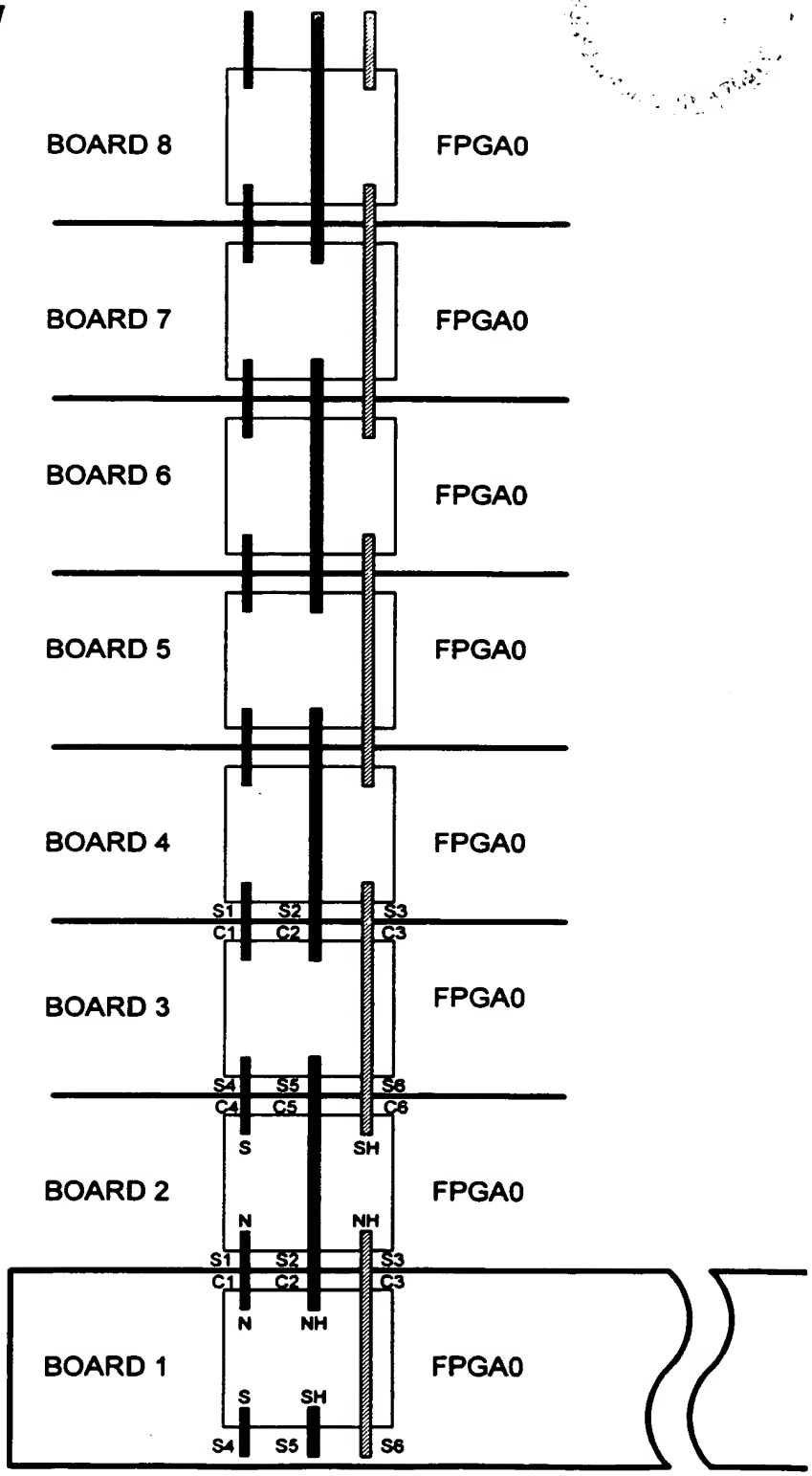


FIG. 85

TWO-ROW FPGA PER BOARD

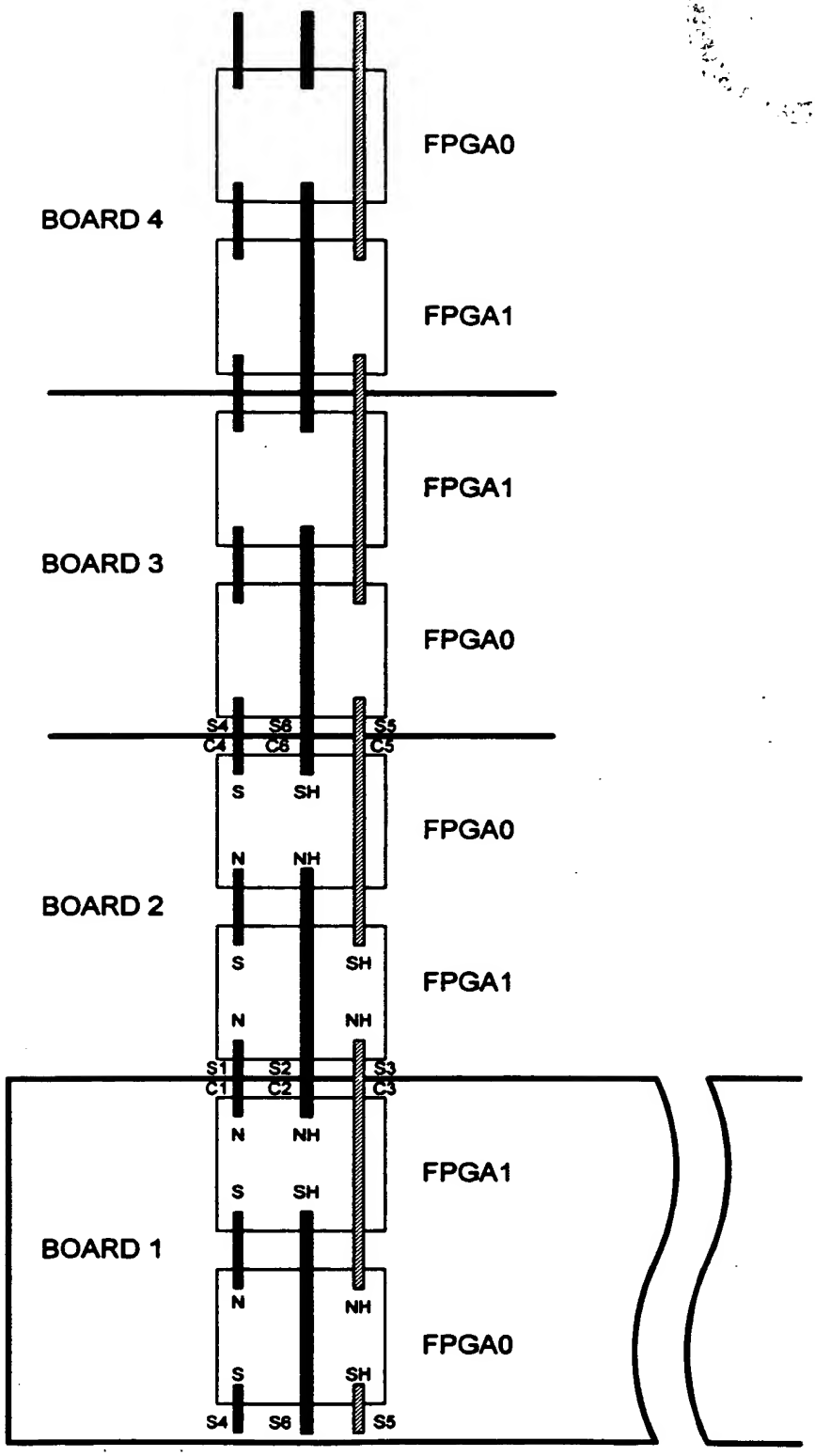


FIG. 86

THREE-ROW FPGA PER BOARD

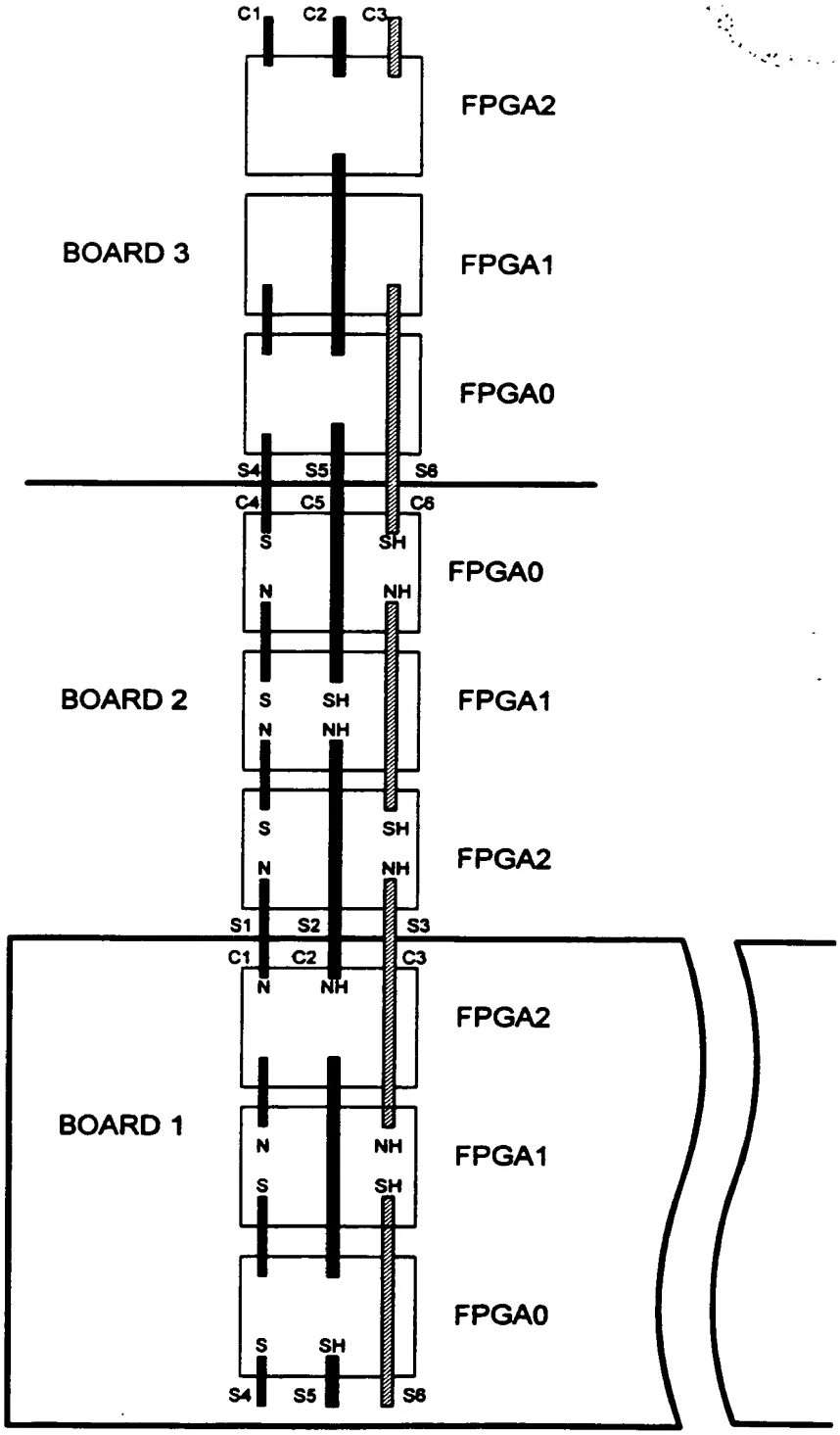


FIG. 87

FOUR-ROW FPGA PER BOARD

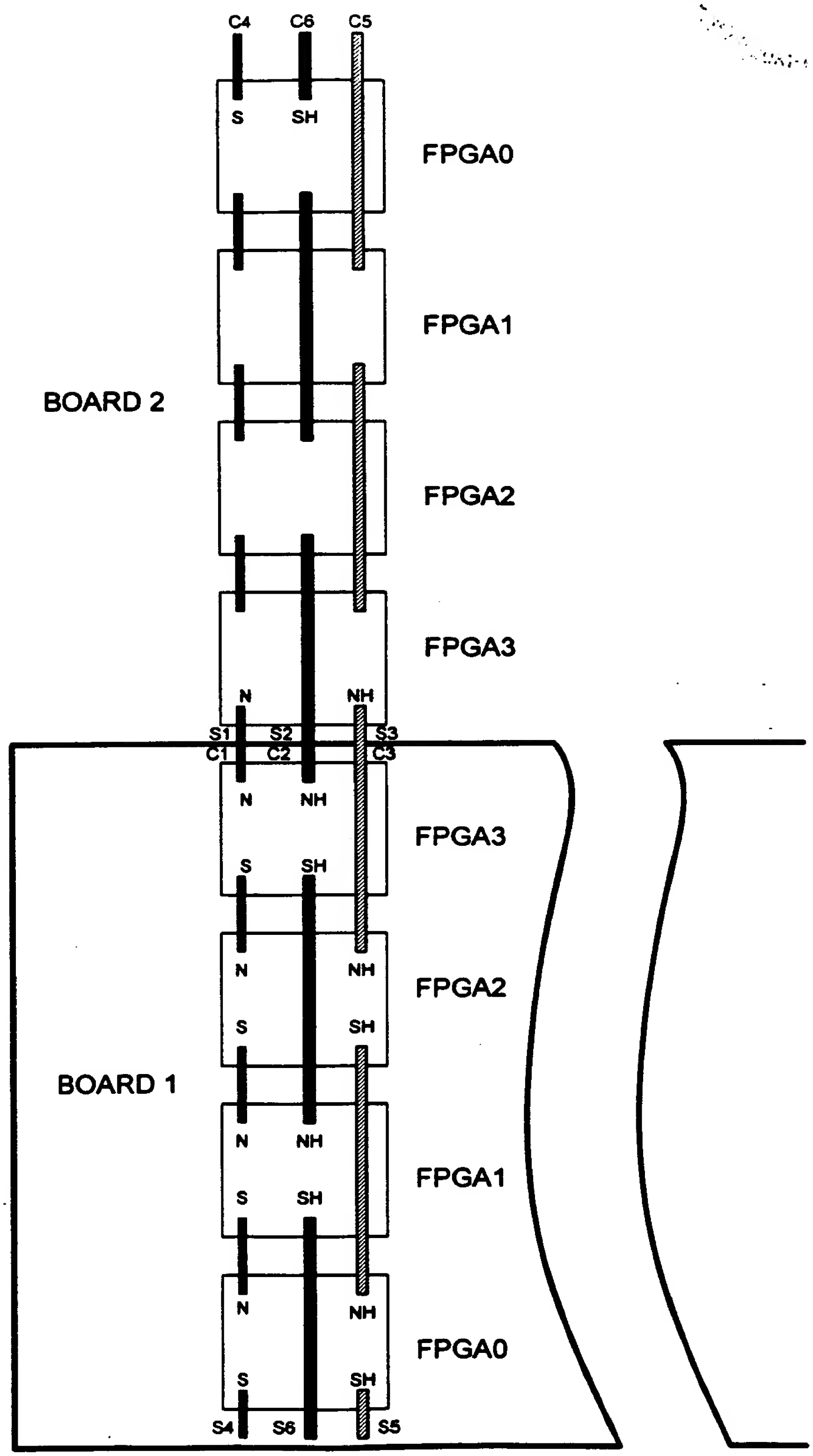


FIG. 88

INTERCONNECT FOR THREE-ROW PER BOARD

I/O Signals	Odd Board	Even Board	Common Board
	Connector-Group Pin-position	Connector-Group Pin-position	Connector-Group Pin-position
FPGA2_N	C1	S1	C1, S1
FPGA2_NH	C2	S3	C2, S3
FPGA1_NH	C3	S2	C3, S2
FPGA0_S	S4	C4	C4, S4
FPGA0_SH	S5	C6	C6, S5
FPGA1_SH	S6	C5	C5, S6

FIG. 89

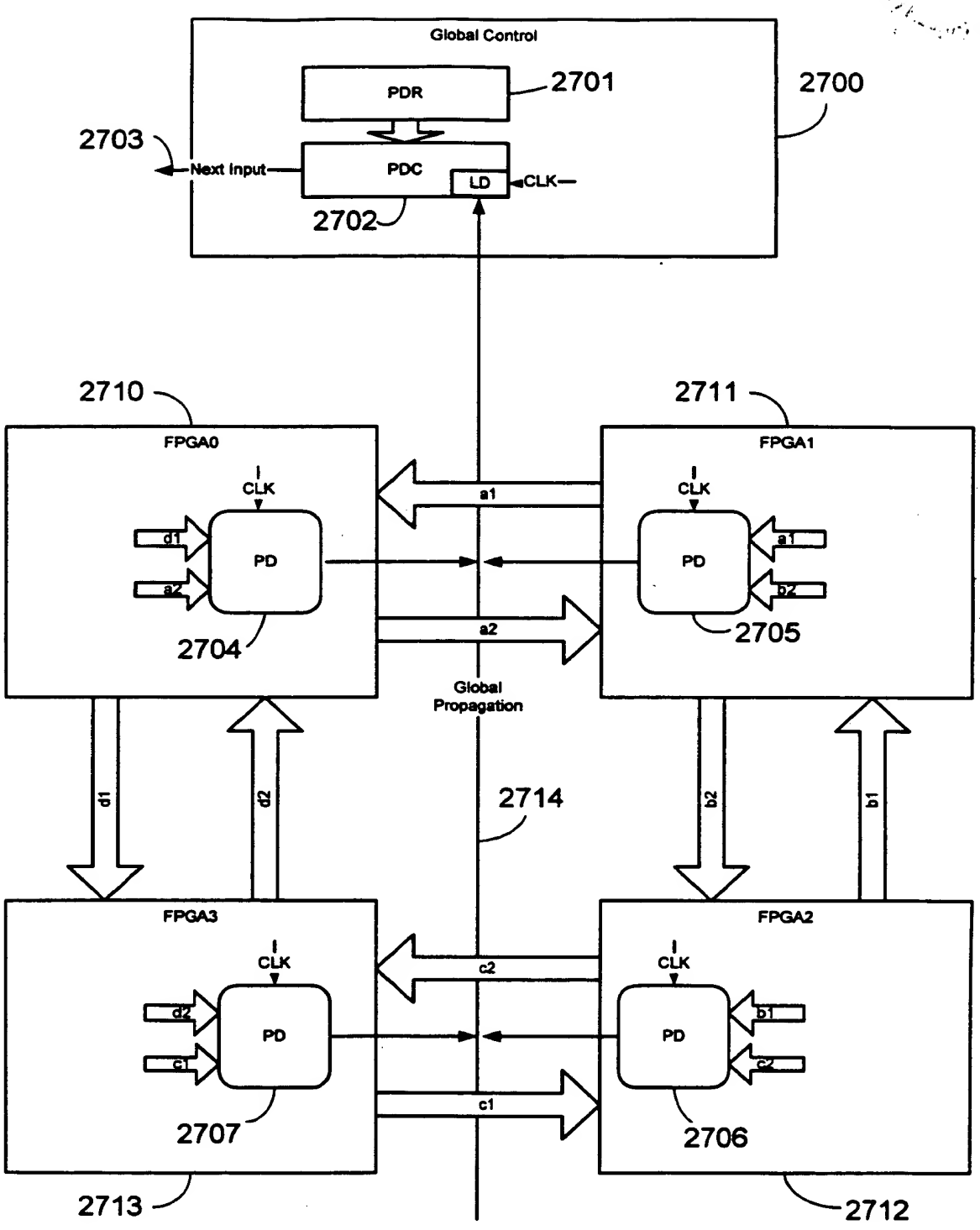

Global Control
PDR
2701
2700
2703
Next Input
PDC
LD
CLK
2702
2710
2711
FPGA0
FPGA1
CLK
d1
a2
PD
2704
a1
a2
b2
2705
Global
Propagation
2714
d1
d2
b2
b1
FPGA3
FPGA2
c2
c1
2707
2706
2713
2712

FIG. 90

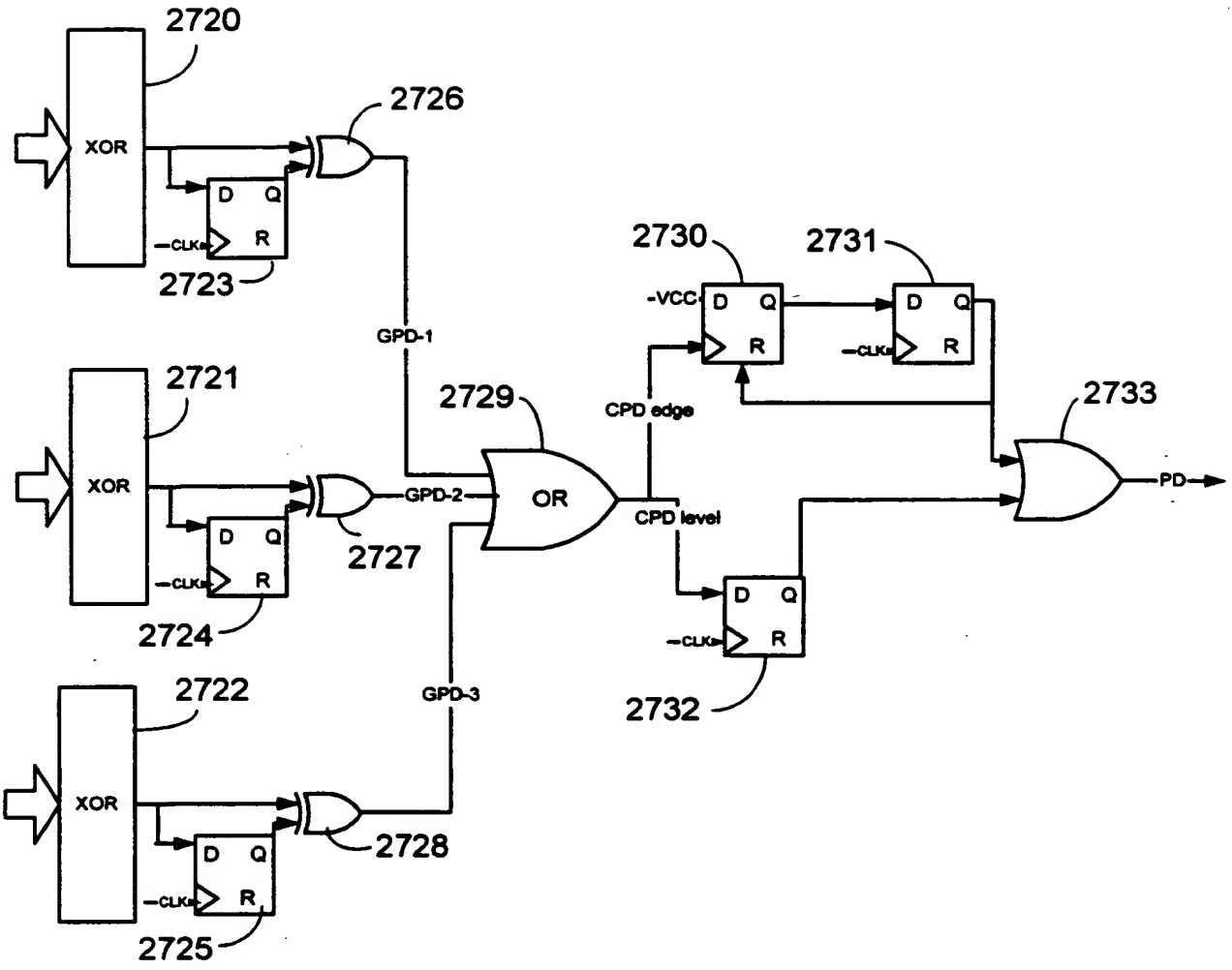

2720
XOR
2726
D Q
R
CLK
2723
GPD-1
2721
XOR
2729
OR
GPD-2
2727
2724
2730
2731
-VCC
CPD edge
2733
PD
CPD level
2732
2722
XOR
GPD-3
2728
2725

FIG. 91

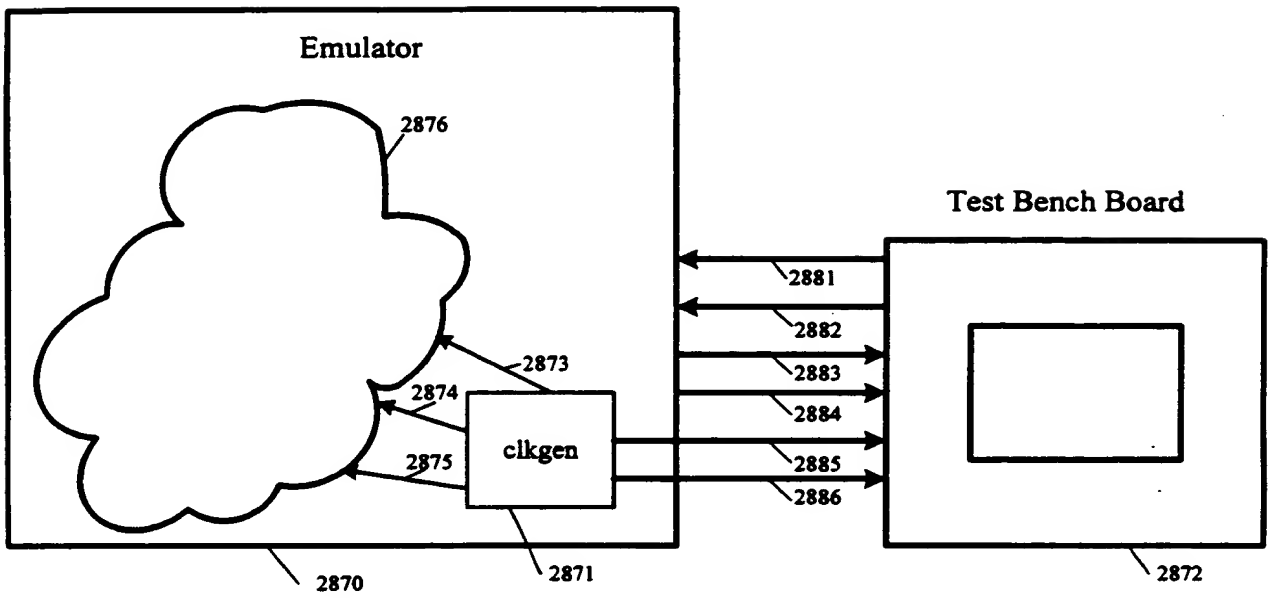

Emulator
2876
Test Bench Board
2881
2882
2883
2884
2885
2886
2873
2874
2875
clkgen
2870
2871
2872

FIG. 92

Clock Specification

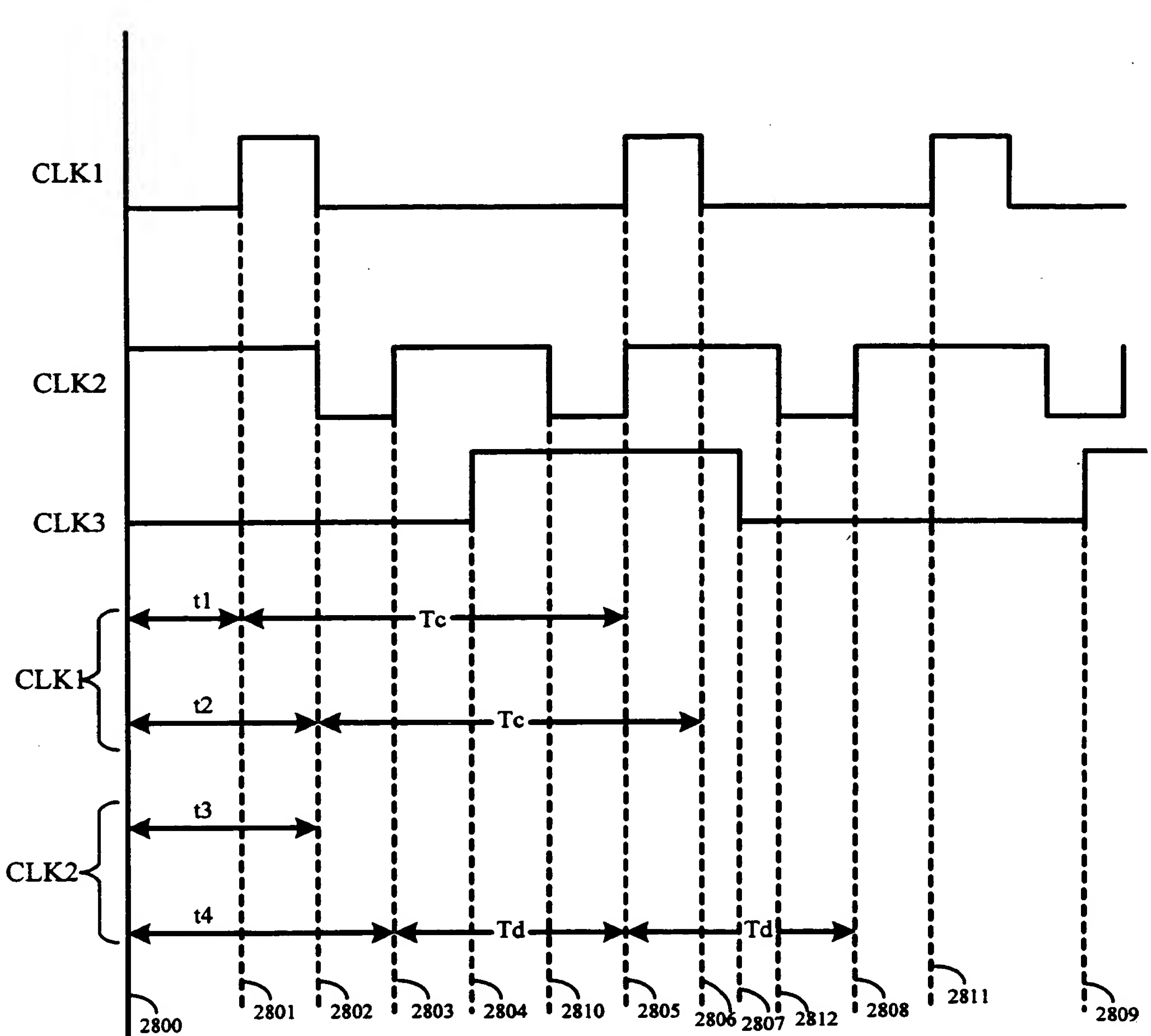


FIG. 93

Clock Generation Scheduler w/ Slices

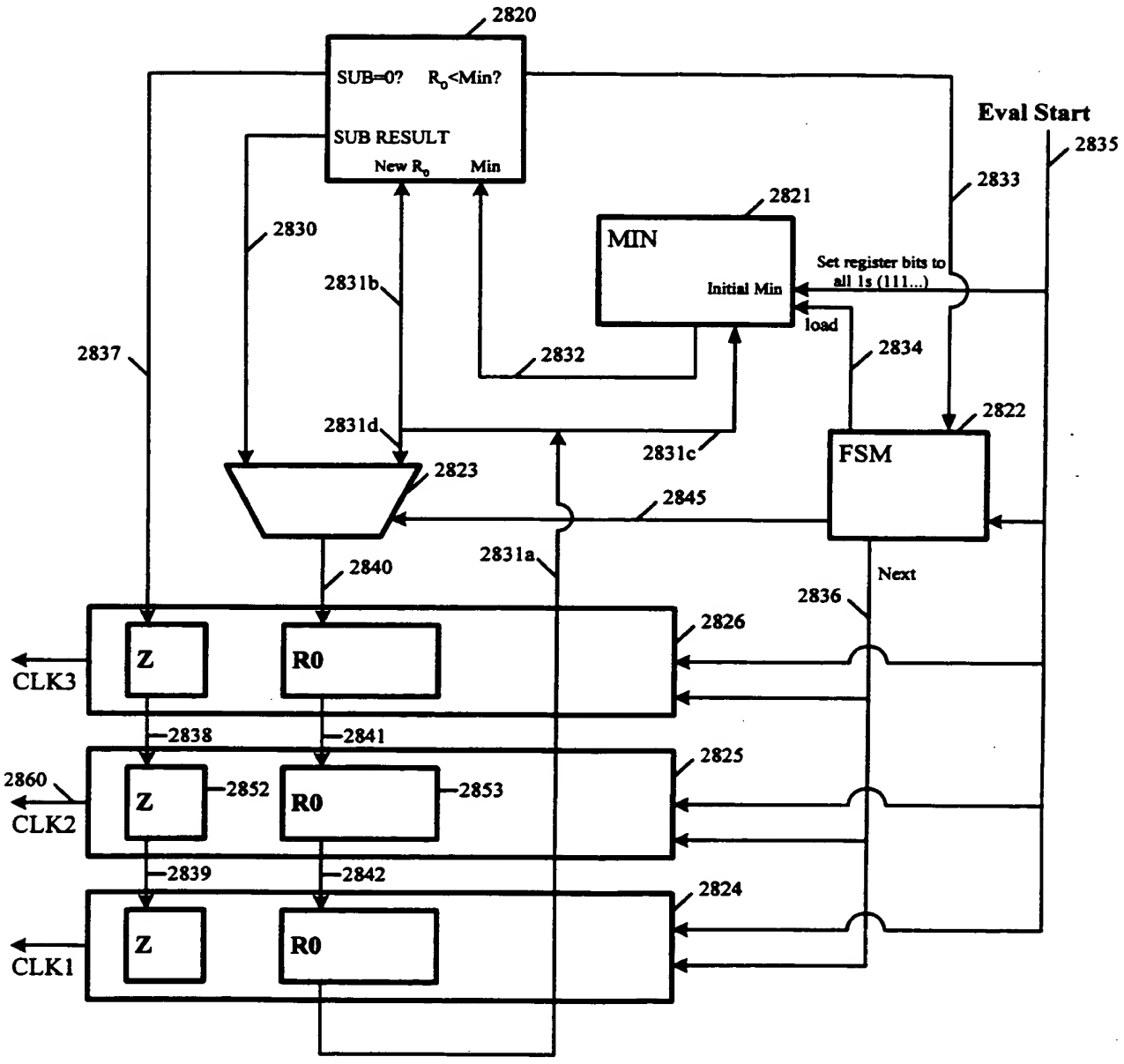


FIG. 94

Clock Generation Slice

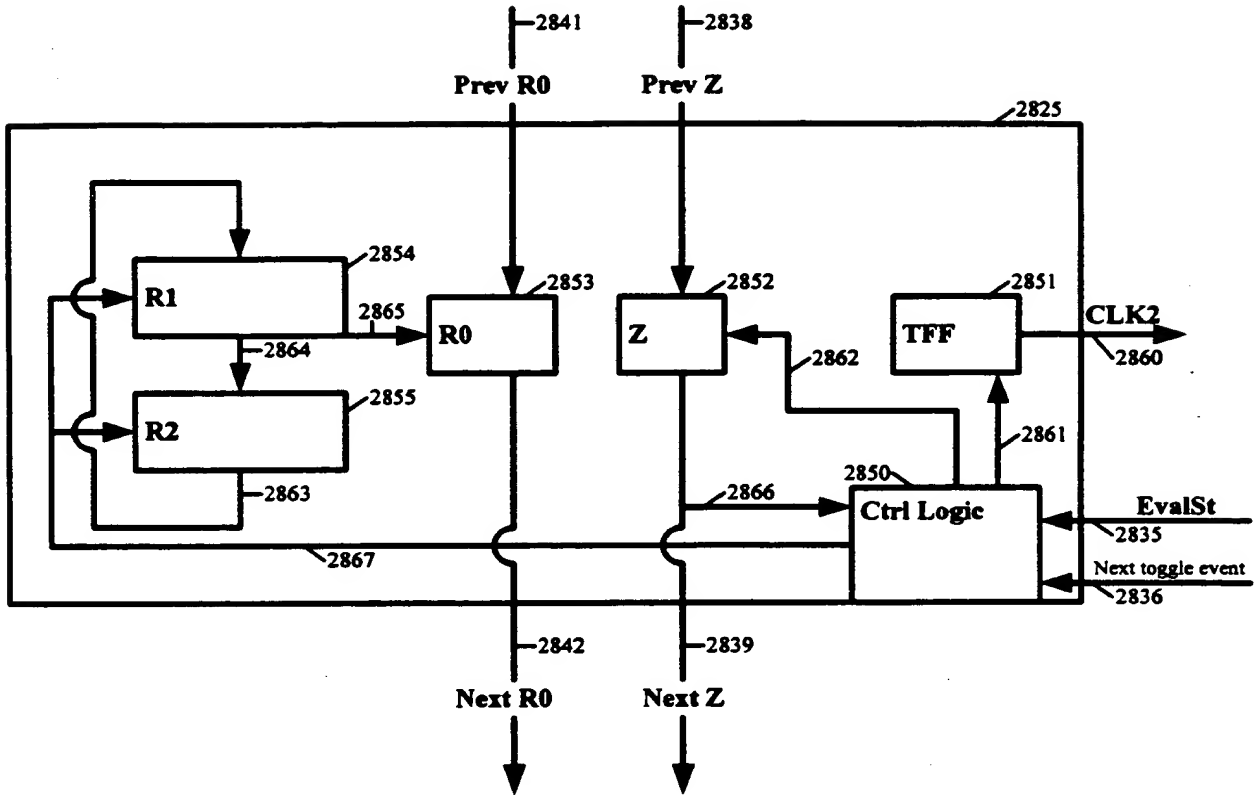


FIG. 95

Clock Generation Scheduler and Slices

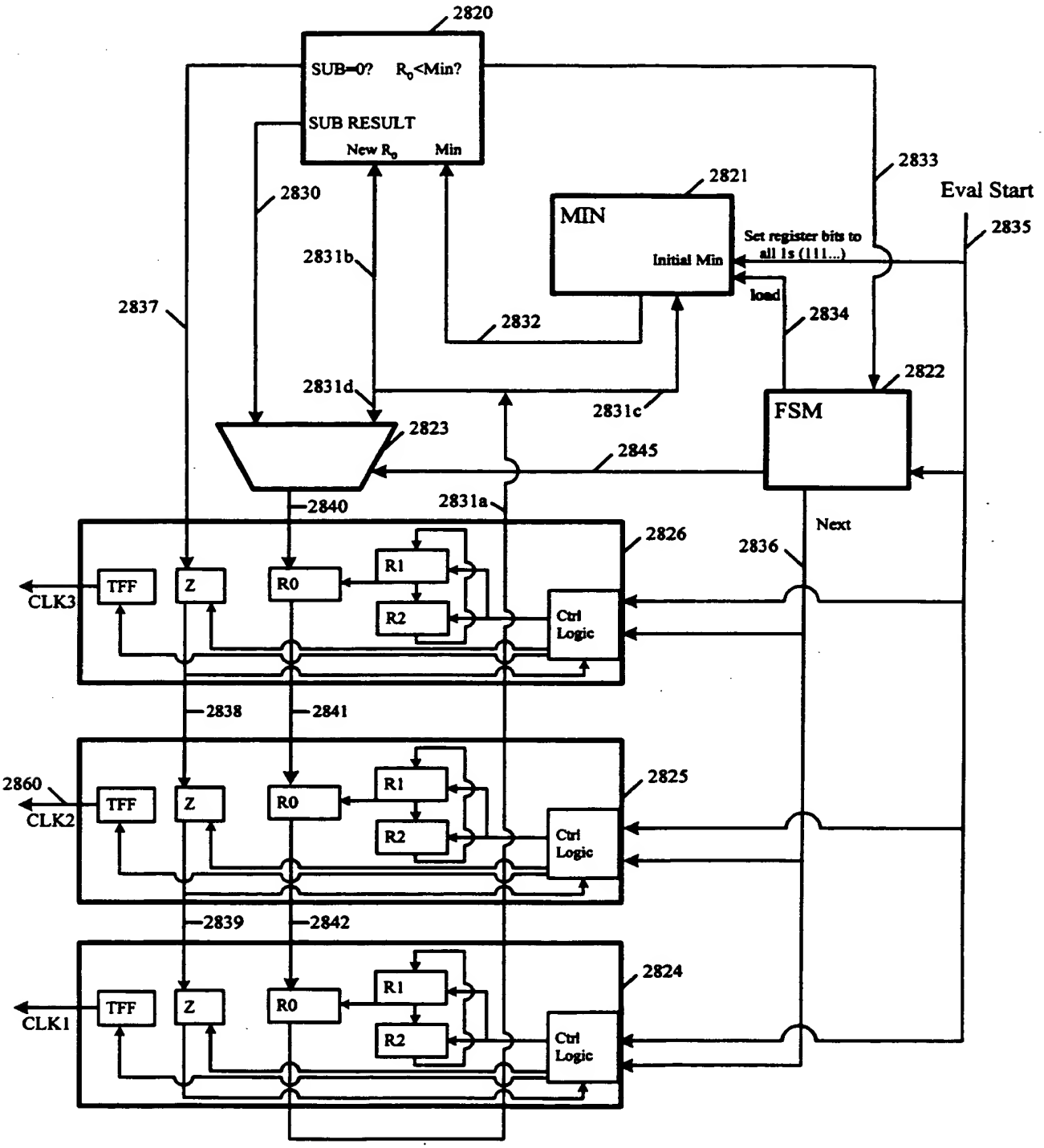


FIG. 96

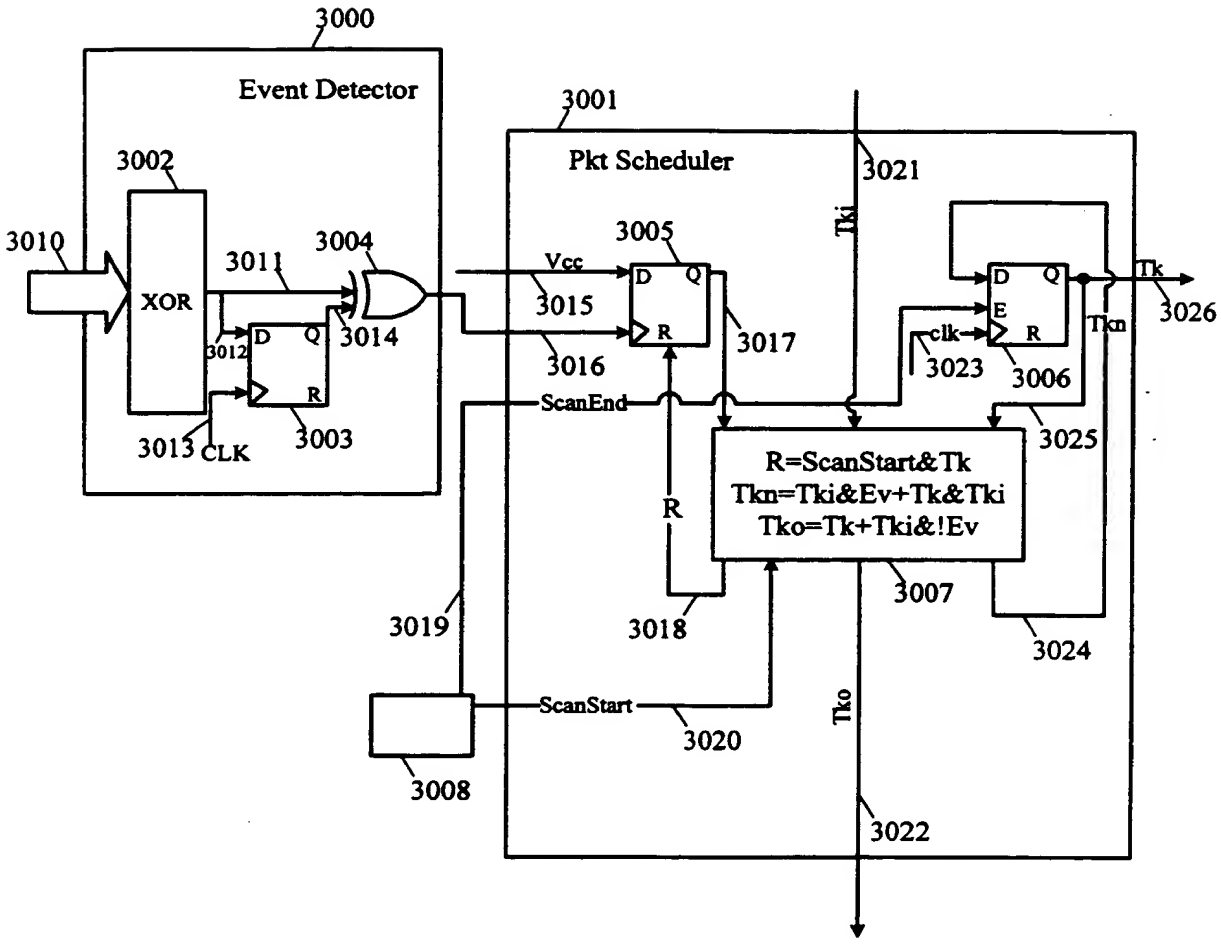

3000
Event Detector
3001
Pkt Scheduler
3002
3010
XOR
3011
3004
3012
D
Q
3014
R
3003
3013
CLK
3021
Tki
3005
Vcc
3015
3016
3017
ScanEnd
clk
3023
E
3006
Tkn
Tk
3026
3025
R=ScanStart&Tk
Tkn=Tki&Ev+Tk&Tki
Tko=Tk+Tki&!Ev
3007
3024
R
3018
3019
ScanStart
3020
3008
Tko
3022

FIG. 97

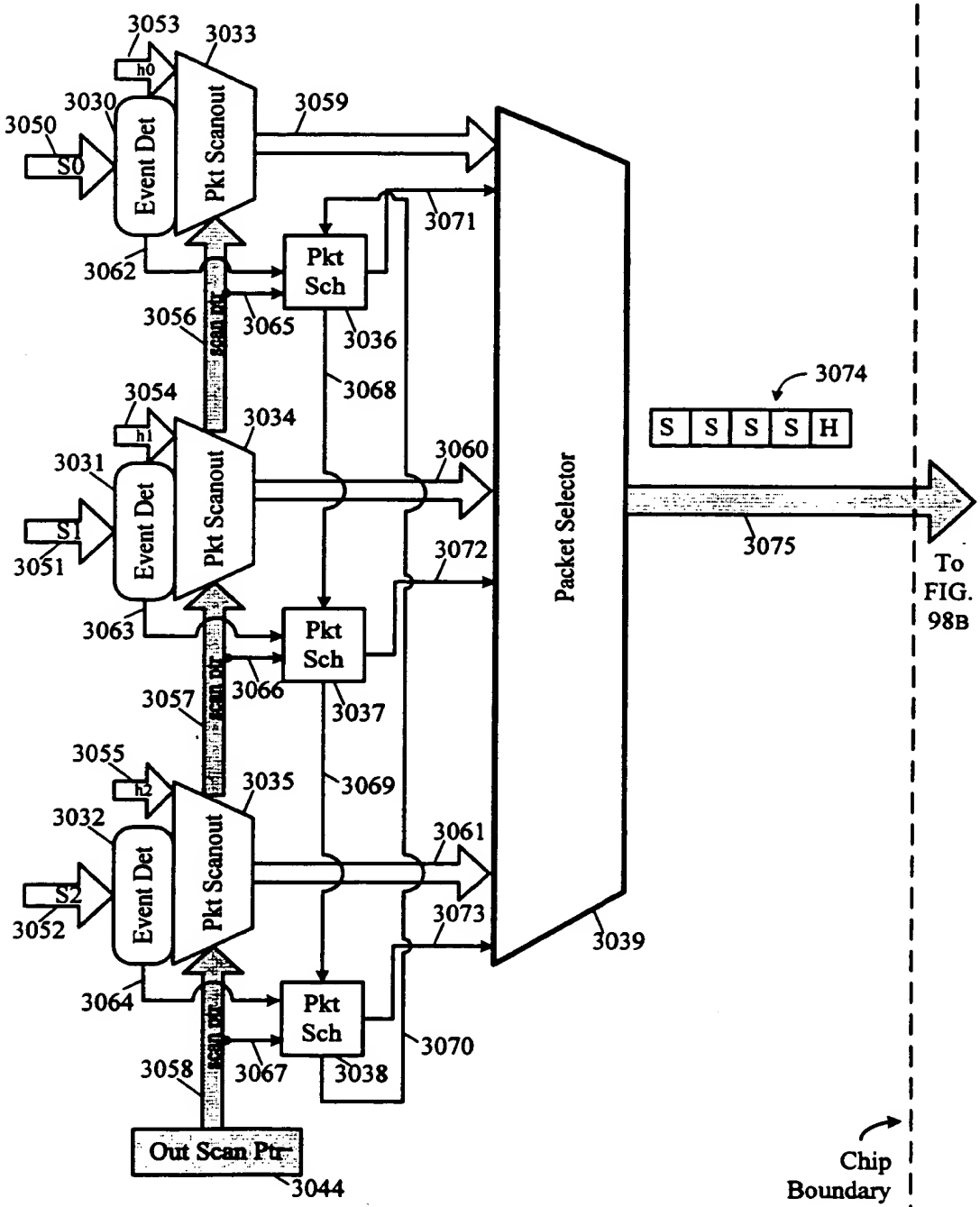


FIG. 98A

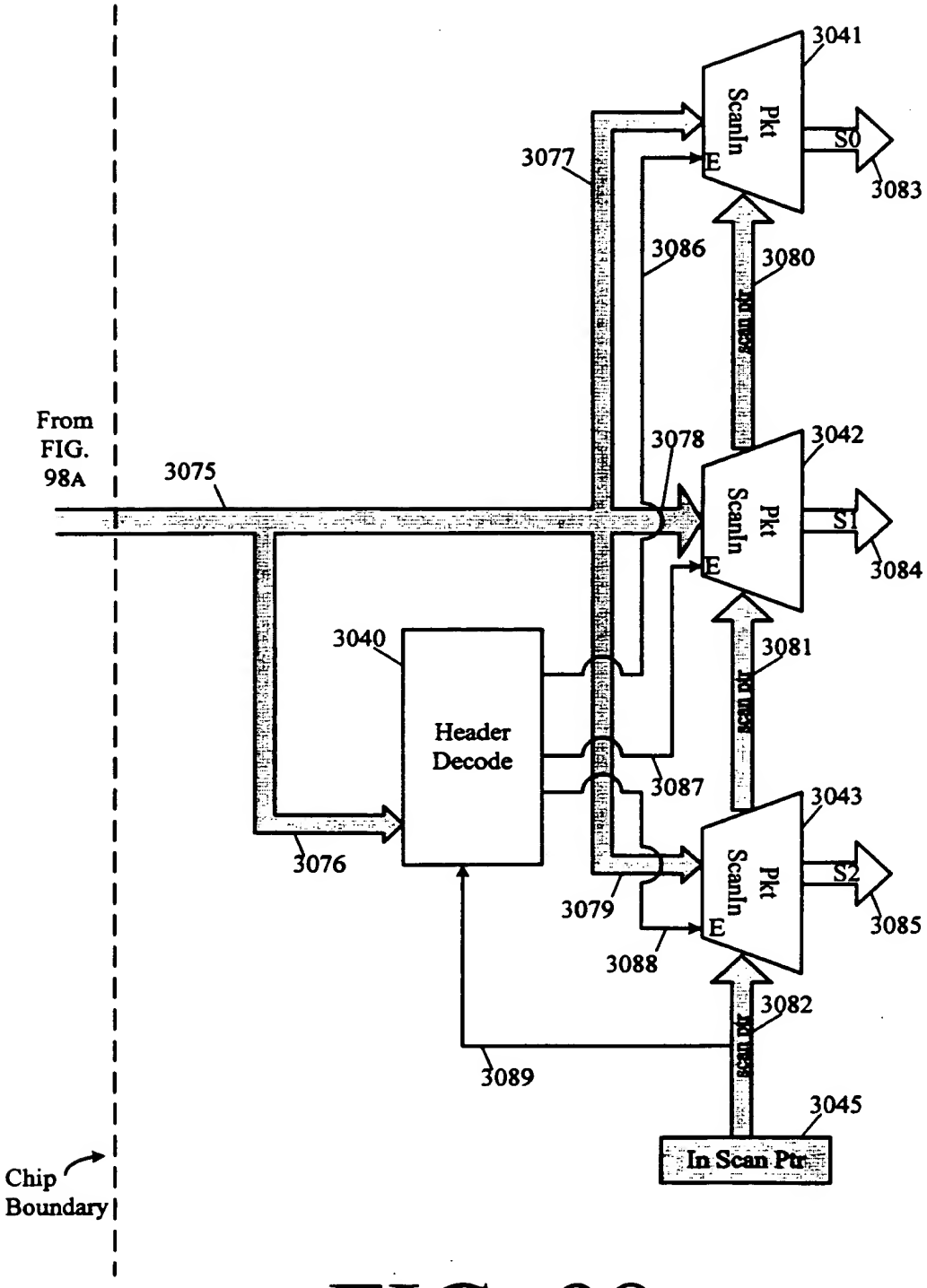

From FIG. 98A
3075
3077
3078
3086
3080
3041
Pkt ScanIn
S0
3083
3042
Pkt ScanIn
S1
3084
3081
3040
Header Decode
3087
3076
3079
3088
3043
Pkt ScanIn
S2
3085
3082
3089
3045
In Scan Ptr
Chip Boundary

FIG. 98B